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. Redfield^f, Stephen Balakirsky^g, Veera Ragavan^h, Howard Liⁱ, 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 Research Laboratory, USA*

^g*Robotics and Autonomous Systems Division, 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, which defines — in natural language — important terms in the domain of

*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. Redfield), `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)

Preprint submitted to Robotics and Computer-Integrated Manufacturing August 13, 2014

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 provide more ontologically sound representations of autonomy and 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 and automation (R&A) is a crucial requirement not only for unambiguous communication and reasoning for robots, but also for knowledge and information sharing about robots among humans and for interaction between robots and humans. 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 mutual agreement among stakeholders, increase the potential for reuse of the knowledge, and promote data integration.

In order to specify and clarify the meaning of the core notions common in R&A, the Working Group (WG) *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 and robot-human communication, robot design, and integration of data about robots. The aim of the ORA WG is to standardize knowledge representation in the 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 theories of the discipline of Formal Ontology [3]. In particular, many of our ontological choices were evaluated based on guidelines from known methodologies, such as METHONTOLOGY [4] and OntoClean [5]. Besides that, CORA was developed based on SUMO [6]; a top-level ontology that aims to define the main ontological categories describing the world. Such an approach is new in developing stan-

30 dards in R&A and has the advantage of producing a better founded standard,
31 which requires less work to use, maintain and extend.

32 This work reports the recent developments within the ongoing CORA
33 project, and provides an overview of its current state. The prior version of
34 CORA [7] has been extended, implementing changes in modeling decisions
35 and introducing new concepts and relations. Thus, this paper presents some
36 changes in modelling decisions that have been implemented since the pre-
37 vious version. The major new contributions can be divided into two broad
38 areas. First, we propose CORAX, an ontology that covers concepts too gen-
39 eral to be part of CORA, and that are not covered by SUMO. These include
40 knowledge about *design* (as in the case of product design), *physical environ-*
41 *ment*, *interaction*, and *artificial systems*. Second, we propose extensions and
42 changes to CORA itself, in order to improve its ontological commitment to
43 the domain. We are primarily concerned with representation of *operation*
44 *modes* and *robot parts*. Finally, we discuss some directions regarding new,
45 yet to be covered topics (such as control and planning).

46 2. Ontology Engineering

47 We developed CORA using several ontology tools and frameworks. The
48 main methodology is based on METHONTOLOGY [4], which supports the
49 development of ontologies either from scratch, by reuse, or by re-engineering
50 existing ones. It consists of a set of guidelines about how to carry out the
51 activities identified in the ontology development process, the kinds of tech-
52 niques that are the most appropriate for each activity, and the resulting
53 products.

54 We also based many of the underlying *ontological commitments* on *On-*
55 *toClean* [5]. Ontoclean is a methodology for validating the ontological ade-
56 quacy of taxonomic relationships, based on highly generic ontological notions
57 drawn from philosophy, like *essence*, *identity* and *unity*. These notions are
58 used to characterize relevant aspects of the intended meaning of the proper-
59 ties, classes, and relations that compose an ontology. OntoClean requires the
60 ontology engineer to explicitly identify the ontological commitments under-
61 lying the concepts that are being modelled. As a result, OntoClean allowed
62 us to identify ambiguities in the definitions of core notions provided by other
63 standards of R&A (see [7] for more details).

64 In addition, as a result of an evaluation process carried out in [7], we

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

72 SUMO defines the basic ontological categories across all domains. The re-
 73 mainder of this section gives a brief overview of its main concepts, illustrated
 74 in Fig. 1. Detailed information can be found in [6].

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

¹<http://www.ontologyportal.org/>

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 has 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 the essential parts that define its identity. A good analogy is to think that perdurantists see objects as tunnel-like regions in a 4D space, while endurantists see them as a 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.

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 the details of each concept, since they were presented in [7]. Instead, we provide a short description of each domain entity.

The term *robot* may have as many definitions as there are people writing about the subject. This inherent ambiguity in the term might be an issue when specifying an ontology for a broad community. We, however, acknowledge this ambiguity as an intrinsic feature of the domain, and therefore have decided to use a definition based purely on necessary conditions, without specifying sufficient conditions. Thus, our goal is to ensure that CORA’s definition of robot includes most of the entities that the community actually considers as robots, at the cost of classifying as robots some entities that actually would not be considered as robots in the point of view of

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

111 some roboticists. However, the concepts in our ontology could be extended
 112 according to the needs of specific sub-domains or applications of R&A.

113 More importantly, our definition of robot emphasizes its functional as-
 114 pects. For our general purposes, *robots are agentive devices* in a broad sense,
 115 designed to perform purposeful actions in order to accomplish a task. In
 116 some cases, the actions of a robot might be subordinated to actions of other
 117 agents, such as software agents (bots) or humans. Robots are also *devices*,
 118 composed of suitable mechanical and electronic parts. Robots can form *social*
 119 *groups*, where they interact to achieve a common goal. A robot (or a group
 120 of robots) can be combined with other devices to form robotic systems. An
 121 environment equipped with a robotic system is a robotic environment.

122 A *robot* is a *device* in the sense of SUMO. According to SUMO, a device
 123 is an artifact (i.e., a *physical object product of making*), which participates

124 as a tool in a process. Being a device, robot inherits from SUMO the notion
125 that devices have parts. Therefore, CORA allows one to represent complex
126 robots with robot parts.

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

135 *Robotic systems* are systems composed of robots (or robot groups) and
136 other devices that facilitate the operations of robots. A good example of a
137 robotic system is a car assembly cell at a manufacturing site. The environ-
138 ment is equipped with actuated structures that manipulate the car body in
139 a way that the industrial robots within the system can act on it. Finally,
140 as previously stated, an environment equipped with a robotic system is a
141 *robotic environment*. See [7, 8] for a more detailed discussion on CORA’s
142 main concepts. Next, we describe new notions that have been integrated
143 into CORA.

144 4. Updating CORA

145 CORA has been updated since its initial proposal in [7, 8]. The main
146 driving force behind these changes came from aligning it with existing on-
147 tologies and more expert involvement in the development process. We com-
148 pared CORA with an *ontology for kitting* developed within the group [9]. This
149 enabled us to investigate whether or not both ontologies could be merged,
150 and to check whether all notions in the kitting ontology were represented in
151 the combination of SUMO and CORA. We found that important concepts
152 and relations present in the kitting ontology were not covered. Due to this,
153 we developed new ontology modules to bridge the gap between SUMO and
154 the kitting ontology, which are mostly covered by CORAX and the POS
155 ontologies.

156 Furthermore, after the preliminary draft standard was completed, we ex-
157 perience increased involvement of independent experts and received addi-
158 tional feedback. Apparently, experts were more comfortable discussing con-
159 cepts and relations, after a first set of ontological commitments were made

160 and the scope of the project was established. The initial model served as a
 161 reference to articulate new requirements on the ontology. Since the initial
 162 model was based on well-founded ontological commitments, the model was
 163 more resilient to ad-hoc proposals to change it, translating into a more stable
 164 evolution of the ontology. Notably, changes were more prominent in aspects
 165 of the ontology that had a less solid foundation in the first version of the
 166 ontology, such as autonomy.

167 In the following sections, we describe the changes made in and around
 168 CORA as a result of that process. They consist mostly of sub-ontologies
 169 complementing or extending 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.

170 5. CORAX: connecting CORA and SUMO

171 Naturally, SUMO does not cover every possible aspect of reality, even
 172 when we restrict ourselves to R&A. At the same time, some of parts of reality
 173 are too general to be included in CORA. We introduced the CORAX ontol-
 174 ogy to address this problem by bridging SUMO and CORA. In particular,
 175 CORAX includes concepts and relations associated with design, interaction,
 176 and environment, which are not covered in SUMO.

177 5.1. *Design*

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

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

189 From our point of view, SUMO does not provide a good specification of
190 design. One of its sub-ontologies—namely the engineering ontology—defines
191 the concept *Model*, which is an abstract entity that seems to capture the
192 notion of design described above. However, a model is not clearly related to
193 content bearing objects, or to *artifacts* in general. SUMO defines a relation-
194 ship called *models*, which is held between *Model* and *Engineering Component*.
195 However, this relationship is too restrictive for our purposes, since we would
196 like to represent models of any kind of artifact.

197 In response to this, we defined the concept of *Design*, which is a kind
198 of *Proposition*. According to SUMO, a *proposition* is an abstract entity
199 that expresses a complete thought or a set of thoughts. For instance, the
200 phrases “*the cat is on the mat*” and “*o gato está no tapete*” express the *same*
201 *proposition* in English and in Portuguese, respectively. In much the same
202 way, different *blueprints* might express the same *design*.

203 Furthermore, the properties of the object must be expressed in its design.
204 For instance, the design of a phone is about an ideal (*idealized*) phone that
205 is materialized in the individual realizations of the design. This ideal phone
206 has ideal properties, such as ideal weight and shape. There are many ways of
207 representing an idealized object within an ontology. For instance, one could
208 represent it as a special instance of the concept *Phone*, called prototype.
209 Another alternative is to collapse both the design and the ideal object into
210 the same entity. This is exactly the approach that was adopted in the design
211 ontology that is presented in [10], which is also based on SUMO. However,
212 since the ideal object is also a proposition, there might be issues when mod-
213 elling its attributes and parts. For instance, if both the design of a phone and
214 the ideal phone (the content of the design) are the same entity, this entity,

Figure 4: Entities associated with Design in CORAX.

215 as a proposition, will have a designed color and a designed shape. However,
 216 *a proposition cannot have a color or a shape*. Thus, we model the ideal ob-
 217 ject as a separate abstract entity called *Design Object*, which specifies the
 218 idealized object that is the *content* of a Design. We believe this definition
 219 better matches the experts' intuitive notion of an engineering model; it also
 220 eliminates the need for a new metacategory in SUMO (such as prototype).
 221 As with physical objects, design objects have properties such as weight and
 222 shape. SUMO provides two main relations to represent properties, namely
 223 *attribute* and *measure*, but these can only predicate physical objects. We
 224 therefore created the relations *designAttribute* and *designMeasure*, which are
 225 analog to attribute and measure in SUMO, allowing the reuse of their domain
 226 values. In this way, we can specify that, for instance, an idealized phone (an
 227 instance of *Object Design*) has a *design shape* and a *design weight*.

228 Designs *idealize* artifacts (therefore, the relation *CORAX:idealizes* in Fig-
 229 ure 4). It is important to note that it is the *design* that idealizes the artifact,
 230 and not the design object. The properties of the design object and those of
 231 the artifact may correlate, but we will not provide a theory about how this
 232 correlation occurs at this stage.

233 5.2. *Physical Environment*

234 Another important notion missing in SUMO is that of *physical environ-*
 235 *ment*. We added this concept to CORAX in order to support specification of
 236 *robotic environments*. In our view, an *environment* is intuitively composed of
 237 a physical region, plus other eventual physical entities that characterize the
 238 environment. In addition, the definition of physical environment depends on
 239 the presence of a landmark (another physical entity) from which it is possible
 240 to define the main region of an environment. Landmarks may or may not
 241 be located within the region of interest of the environment. For instance, an
 242 office room environment depends on the physical configuration of its walls,
 243 which are located in the environment. But we can also define an arbitrary
 244 environment consisting of a cube in outer space that depends on Earth as a
 245 landmark. In this case, Earth does not need to be located within or at the
 246 borders of the region.

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

247 More formally, we define a physical environment in CORAX as a physical
 248 object that has at least one region as *part* and that depends on another entity.
 249 All other physical objects that are part of an environment must be located
 250 within a region that is part of the environment.

251 5.3. *Interaction and Artificial Systems*

252 In order to properly define a *robotic system*, we have to specify what is
 253 an *artificial system*. An artificial system is simply an artifact formed from

254 various devices and other objects that interact with each other and with the
255 environment in order to fulfill a function.

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

262 6. CORA: Autonomy revisited

263 *Autonomy* is one of the most important terms in R&A, yet one of the
264 hardest to define precisely. In the previous version of CORA, we advocated
265 for a flexible definition that — while not being precise — could distinguish
266 between robots that were clearly autonomous from others with questionable
267 autonomy. In CORA, it has now been pushed a step further in order to make
268 the modelling more versatile.

269 In this new version, our definitions are aligned with those from the ALFUS
270 [11] framework, which was the result of an extensive study on autonomy
271 in unmanned vehicles. In short, ALFUS states that autonomy is generally
272 dependent on the *degree of human intervention* and *context*, where the latter
273 is characterized by *type of mission* and *environment*.

274 CORA’s definition of autonomy is closely related to what ALFUS defines
275 *modes of operation for unmanned systems*. These modes stretch from fully
276 autonomous to remote controlled, representing the degree of human interac-
277 tion needed for the robot to perform its task. In our view, they encapsulate
278 the experts’ intuitive notion of autonomy in R&A². More specifically, CORA
279 includes:

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

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

²ALFUS 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.

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

286 ***Teleoperated robot:*** A role for a robot performing a given task, in which
287 a human operator either directly controls the actuators using sen-
288 sory feedback, or assigns incremental goals on a continuous basis. A
289 teleoperated robot will complete its last command after the operator
290 stops sending commands, even if that command is complex and time-
291 consuming.

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

297 ***Automated robot:*** A role for a robot performing a given task, in which the
298 robot acts as an automaton, following pre-defined (scripted) plans, not
299 adapting to changes in the environment.

300 It is important to note that *automated robot* is not part of ALFUS' modes
301 of operation. Experts in our groups determined that certain robots require
302 little human interaction, but at the same time are too simple to be charac-
303 terized as autonomous. This is the case of automatons, including automated
304 dolls and toys, which *cannot* react to changes in environment. Relatively
305 simple code scripts or mechatronics determine the behavior of these robots.

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

313 The fact that this classification of autonomy is context-dependent also
314 affected our modelling choices. In a modelling sense, a mode of operation is
315 a *role*. A role can predicate a given entity at a given time, but it can cease
316 to predicate it at a later time. For instance, the canonical example of role is
317 *Student*: one can predicate a person as a student at a given time, and later
318 cease to do so. This contrasts with rigid types, such as *Person*. Someone

cannot cease to be a person without ceasing to exist. In general, a role is also dependent on another entity. For instance, a person must 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 [3]). 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 *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* 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 operational mode depends on the way a robot determines the outcome of the processes it is involved in. We represent the operational modes as subrelations of *robotAgent*: *fullyAutonomousRobot*, *semiAutonomousRobot*, *teleoperatedRobot*, *remoteControlledRobot* and *automatedRobot*. When a particular robot assumes a particular operational mode for a particular task, it is predicated with the appropriate relation. For instance, a robot that can drive autonomously, assumes the role *fullyAutonomousRobot* for the autonomous driving process. The same robot can assume different operational modes in different processes, depending on the context. Interestingly, since processes can have sub-processes, a robot can assume different roles for different sub-processes. For instance, a cleaning robot might be fully autonomous as it detects dirty places to clean, but simultaneously be semi-autonomous with respect to planning routes around the house, or vice versa.

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. A myriad of devices 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 designs are only constrained by human

355 needs, human creativity and available technological resources. Therefore, a
356 type of device that has never been considered as a potential robot part can
357 be used as a robot part by some future designer. An ontology for R&A, as
358 CORA is, must take this issue into account.

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

371 In the earlier version of CORA [7], the notion of robot part was considered
372 as a *class*, whose instances are not *essentially* instances of it. Thus, instances
373 of robot part could cease to be robot parts, without ceasing to exist. In this
374 sense, for example, an instance of power source that is considered as a robot
375 part at a given moment (when it is connected to a robot) could cease to be
376 a robot part in another moment without ceasing to exist (as an instance of
377 power source). Thus, *Robot part* was considered as an *anti-rigid* class, in the
378 sense of [5, 3]. Our modelling pattern [7] was developed accordingly, inspired
379 by [3]. It represents how a specific instance of a specific kind of device (e.g.,
380 power source) could be classified as a robot part.

381 This pattern becomes complex when we take into account the principles
382 advocated in [5, 3]. According to these frameworks, an anti-rigid class (e.g.,
383 robot part) cannot subsume a rigid one (e.g., power source). Considering
384 this principle, for each rigid class c that can play the role of robot part,
385 we must create another specific anti-rigid class (a specific role) that will be
386 subsumed by both c and *Robot Part*. For example, an instance of the rigid
387 class *Wheel* only becomes a robot part when it is attached to a particular
388 robot. Given this condition, it becomes a member of the more specific class
389 (e.g., “*Wheel as Robot Part*”), which is subsumed by the rigid class *Wheel*
390 and the anti-rigid class *Robot Part* (see [7] for further details.)

391 The representation of robot parts in the new edition of CORA was changed,
392 mainly because the modelling pattern proposed for representing robot parts

393 results in domain models that are overwhelmingly complex. Some classes
 394 that must be created in order to maintain the consistency of the model do
 395 not fit well into the domain conceptualization, and the resulting complex-
 396 ity is hard to manage. Therefore, this modelling pattern could hinder the
 397 broad adoption of the ontology in the domain. Another factor leading to
 398 the revision was that it is not clear how to fit the dynamical behavior that
 399 is expected from roles in the framework of SUMO. The modelling of roles
 400 adopted in [5, 3] relies on the notion of *possibility* (a *modal* notion). However,
 401 as pointed out in [12], the treatment of possibilities in SUMO is not clear.

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

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

416 ***Robot actuating part:*** responsible for allowing the robot to move and act
 417 in the surrounding environment. Formally, robot actuating parts must
 418 be devices that are instruments in a process of robot motion, which is
 419 any process of movement where the robot is the agent and one of its
 420 parts is acted upon.

421 ***Robot communicating part:*** responsible for providing communication among
 422 robots and humans, by allowing the robot to send (or receive) informa-
 423 tion to (or from) a robot or a human.

424 ***Robot processing part:*** responsible for processing data and information.
 425 Formally, robot processing parts must be processing devices connected
 426 to the robot. A processing device is any electric device whose purpose
 427 is to serve as an instrument in a subclass of computer process.

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

431 This modelling choice also provides interesting modularity characteris-
 432 tics. It keeps CORA as a minimal core of high-level concepts that provide
 433 the structure to the domain without going deep into details regarding the
 434 myriad of different devices that could play the roles specified here. In this
 435 sense, this structure of roles can be viewed as an interface (in the sense of
 436 *object oriented programming paradigm*) that can be implemented in different
 437 ways. Naturally, this schema poses the need for sub-ontologies to define the
 438 taxonomies of devices that can play the roles specified in CORA, such as an
 439 *ontology of sensors, ontology of grippers, etc.*

440 8. POS: Position, orientation and pose

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

449 According to the literature, roboticists and other domain experts usually
 450 utilize two kinds of positional information [13]: *quantitative* and *qualitative*.
 451 In the quantitative case, a position is represented by a *point* in a given
 452 coordinate system. In the qualitative case, a position is represented as a
 453 *region* defined as a function of a reference object. For instance, one can
 454 describe a robot as being positioned at the coordinates (x, y) in the global
 455 coordinate system, or that the robot is positioned *at the front of the box*,
 456 where “front” comprises a conical region centered on the box and pointed
 457 forward.

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

462 *Position measurements* are *physical quantities* that can be *position points*
 463 or *position regions*. A position point refers to a point in a *coordinate system*

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

464 projected on the physical space. A position region is an *abstract region* in a
 465 *coordinate system* defined with reference to a series of position points.

466 A position point denotes the *quantitative* position of an object in a co-
 467 ordinate system. More specifically, position points are always defined in a
 468 single coordinate system.

469 A *coordinate system* is an *abstract* entity that is defined in relation to a
 470 *single reference* object, i.e., there is an object that is the reference for each
 471 coordinate system. For instance, the local coordinate system of a robot is
 472 referenced by the robot itself. Additionally, the reference object does not
 473 need to be at the origin of the coordinate system.

474 This ontology does not commit to a particular kind of coordinate sys-
 475 tem. It can be stated however, that a coordinate system defines at least
 476 one dimension in which points get their coordinate values. An n -dimensional
 477 coordinate system, c , is homeomorphic to a subset of $\mathbb{R}^n$, such that a coor-
 478 dinate $p \in c$ can be represented as n -tuple $\phi(p) = (x_1(p), x_2(p), \dots, x_n(p))$. The
 479 functions $x_1, x_2, \dots, x_n$ are coordinate functions that attribute to p a real

480 value in the dimension n of the coordinate system [14].

481 A fundamental aspect of coordinate systems is the notion of *transformation*,
482 which maps position points in one coordinate system to position points
483 in another coordinate system. Transformations can be composed generating
484 new transformations. In our ontology, an object can display multiple posi-
485 tions in different coordinate systems only if there is a transformation that
486 can map between the two.

487 In addition, coordinate systems are related through *hierarchies* (i.e. trees).
488 We say that a given coordinate system c_1 is parent of a coordinate system
489 c_2 if there is a transformation t_1 that maps the points of c_1 to points in
490 c_2 , and there is a transformation t_2 that maps the points of c_2 to points in
491 c_1 . According to this, if two coordinate systems share a parent node in the
492 hierarchy tree, there is a transformation between them. Usually, an agent
493 chooses a coordinate system as the global reference frame that constitutes
494 the *global coordinate system* (GCS) for that agent. This GCS can be *arbi-*
495 *trarily* chosen and does not have reference to a particular coordinate frame.
496 *Local coordinate systems* (LCS) are defined in relation to GCS by hierarchical
497 links. This hierarchy is arbitrary, in the sense that it can be defined by the
498 designer or agent.

499 As already stated earlier, besides the quantitative position, our ontology
500 also provides concepts about qualitative positions that are defined in terms of
501 position regions. Example of qualitative positions are “left of”, “in front of”,
502 “on top of”, etc. These expressions define regions in relation to a reference
503 object o_r in which other objects are placed. More specifically, a *position*
504 *region* is composed of poses in the coordinate system generated by a *spatial*
505 *operator* on the reference object. The spatial operator is a *mathematical*
506 *function* that maps reference objects to regions in a coordinate system in
507 arbitrary ways.

508 Our ontology also allows for the representation of *relative positions* of
509 objects with respect to a given reference object. In general, this kind of in-
510 formation is represented through *spatial relations* that hold between objects.
511 An example of this kind of information is the relation $\text{leftOf}(o, o_r)$, which
512 represents that the object o is positioned to the left of the object o_r . This
513 kind of relation can be defined in our framework using the notions of *relative*
514 *position* and *spatial operator*. For example, the relation $\text{leftOf}(o, o_r)$ holds
515 when there is a qualitative position s (a position region) that was generated
516 by the spatial operator leftOfOp over the reference object o_r , and the ob-
517 ject o has the relative position s regarding o_r . Through this mechanism, our

518 ontology provides the semantics for spatial relations like “to the left of”.

519 The usual notion of orientation is similar to position as far as its con-
520 ceptual structure is concerned. Due to this, we will provide only a brief
521 overview. An object can have a quantitative orientation defined as a value in
522 an orientation coordinate system, as well as a qualitative orientation defined
523 as a region in relation to a reference object. For example, orientation is used
524 in the phrase “the robot is oriented at 54 degrees”; the orientation value in
525 this case is 54 in the circular, one-dimensional coordinate system of a com-
526 pass. On the other hand, orientation regions capture a less intuitive notion.
527 The expression “the robot is oriented to the north of the Earth” allows for
528 interpretations where the robot has a range of possible orientation points
529 around 0 degrees. Thus, we model “north” as a region (or interval) in the
530 one-dimensional compass coordinate system that overlaps with the general
531 orientational extension of the object.

532 A position and an orientation constitute a pose. The pose of an object
533 is the description of any position and orientation simultaneously applied to
534 the same object. Often, a pose is defined with a position and an orientation
535 referenced to different coordinate systems/reference objects. In addition,
536 since objects can have many different positions and orientation, they can
537 also have many different poses.

538 It is important to note that the current version of the POS ontology is
539 *synchronic*. That is, it considers only facts about a single time point, just
540 like a snapshot in time. One of the future extensions to this ontology will
541 consider dynamic world modelling, eventually producing a *diachronic* version
542 of the POS ontology.

543 9. Discussion

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

551 Several scenarios could take advantage of CORA (and the related ontolo-
552 gies) in R&A. Firstly, CORA can be immediately applied in *meaning nego-*
553 *tiation* among roboticists. That is, our ontologies could be used as *reference*

554 *conceptual models* for ensuring mutual agreement among humans regarding
555 the meaning of concepts of R&A domains.

556 Moreover, used as a *software component*, the ontology can naturally be
557 applied for enhancing communication among (heterogeneous) robots, as well
558 as among robots and humans. For example, a straightforward application
559 for CORA is as a tool developing a *middleware* for communication, ensuring
560 semantic interoperability between the members of a robot group.

561 Our ontologies can be used as *reusable knowledge components* in *knowledge-*
562 *based problem-solving processes*. Using CORA, thus, a robot can apply high-
563 level logical reasoning capabilities, taking advantage of its high-level knowl-
564 edge about the world to decide which action it should perform in order to
565 achieve its goal. In general, robots can use ontologies to support tasks such as
566 *planning* [15, 16, 17] and *navigation* [18]. Other ontologies can also be inte-
567 grated with our ontologies, providing a wide range of concepts and relations
568 that allow richer descriptions of the robot’s world. Such semantic descrip-
569 tions can be used by the robot in perception processes such as [19, 20, 21, 22]
570 for enhancing tasks that require *object recognition* through *visual perception*.
571 These semantic descriptions can be used also for specifying tasks to the robot,
572 as in [23].

573 Furthermore, our ontologies can be used for defining the notions underly-
574 ing *robot programming frameworks*. CORA could provide these frameworks
575 with a conceptual structure that fits the conceptualization that is shared
576 among the roboticists. For instance, an object-oriented programming frame-
577 work for robots based on concepts and relations in CORA would be more
578 easily assimilated by new programmers. In this way, dealing with these
579 frameworks would become more natural for the practitioners of R&A. In ad-
580 dition, our ontologies could define standard *interfaces* for these frameworks,
581 promoting the semantic interoperability among them.

582 CORA can also be used for promoting *data integration* and *semantic in-*
583 *teroperability* among robot databases. This could have positive impacts to
584 the *knowledge management* process of companies that commercialize prod-
585 ucts and components for the R&A field.

586 10. Future work: what should we expect next?

587 CORA and related ontologies still do not cover some important areas in
588 R&A. For instance, *control* still needs to be taken into account. This issue
589 is complex, since it involves other important concepts in robotics, such as

590 perception, planning, and action. CORA should also incorporate information
591 ranging from simple classical controllers — such as proportional-integral-
592 derivative controllers (PID) — to complex non-linear control. In addition, it
593 should also account for different control strategies.

594 The notion of *task* is also important in this domain. Since robots should
595 be able to operate in complex scenarios, task definitions must be clear to
596 allow robots to communicate with each other, other machines, and humans.
597 In this sense, ontologies play a clear role in task specification. CORA must
598 be designed to allow several types of tasks in various environments, e.g.,
599 grasp, move, scan, and so on. Future work will be devoted to the ontological
600 characterization of what kind of entity a task is. For example, we believe
601 that a good starting point is to separate *tasks* from *task executions*. With
602 this distinction, we acknowledge that tasks are *abstract* entities that describe
603 goals to be reached; while tasks executions are *events* composed by *actions*
604 that are performed by robots in the world in order to reach a given goal.
605 Moreover, in future steps it is necessary to identity the basic kinds of tasks
606 that robots usually perform. These task definitions will be the basis of more
607 complex task definitions. CORA must define clearly the interfaces to domain
608 ontologies, like industrial [24] or surgical [25] [26].

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

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

616 Acknowledgment

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

622 References

- 623 [1] Studer, R., Benjamins, V.R., Fensel, D.. Knowledge engineering: Prin-
624 ciples and methods. *Data and Knowledge Engineering* 1998;25(1-2):161–
625 197. URL [http://dx.doi.org/10.1016/S0169-023X\(97\)00056-6](http://dx.doi.org/10.1016/S0169-023X(97)00056-6).
- 626 [2] Schlenoff, C., Prestes, E., Madhavan, R., Gonçalves, P.J.S., Li, H.,
627 Balakirsky, S., et al. An IEEE standard ontology for robotics and
628 automation. In: *Proc. of the 2012 IEEE/RSJ Intl. Conf. on Intelligent*
629 *Robots and Systems*, Vilamoura. 2012, p. 1337–1342.
- 630 [3] Guizzardi, G.. Ontological foundations for structural conceptual mod-
631 els. PhD thesis; University of Twente; The Netherlands; 2005.
- 632 [4] Fernández, M., Gómez-Pérez, A., Juristo, N.. METHONTOLOGY:
633 from ontological art towards ontological engineering. In: *Ontological*
634 *Engineering*; vol. 6 of *AAAI Spring Symposium*. 1997, p. 33–40.
- 635 [5] Guarino, N., Welty, C.A.. An overview of OntoClean. In: Staab, S.,
636 Studer, R., editors. *Handbook on Ontologies*. Intl. Handbooks on Infor-
637 mation Systems; Springer Berlin Heidelberg. ISBN 978-3-540-70999-2,
638 978-3-540-92673-3; 2009, p. 201–220. URL [http://link.springer.](http://link.springer.com/chapter/10.1007/978-3-540-92673-3_9)
639 [com/chapter/10.1007/978-3-540-92673-3_9](http://link.springer.com/chapter/10.1007/978-3-540-92673-3_9).
- 640 [6] Niles, I., Pease, A.. Towards a standard upper ontology. In: *Proceed-*
641 *ings of the international conference on Formal Ontology in Information*
642 *Systems - Volume 2001*. FOIS '01; New York, NY, USA: ACM. ISBN
643 1-58113-377-4; 2001, p. 29. doi:\bibinfo{doi}{10.1145/505168.505170}.
644 URL <http://doi.acm.org/10.1145/505168.505170>.
- 645 [7] Prestes, E., Carbonera, J.L., Rama Fiorini, S., M. Jorge, V.A.,
646 Abel, M., Madhavan, R., et al. Towards a core ontology for robotics
647 and automation. *Robotics and Autonomous Systems* 2013;61(11):1193–
648 1204.
- 649 [8] Carbonera, J.L., Fiorini, S.R., Prestes, E., Jorge, V.A., Abel, M.,
650 Madhavan, R., et al. Defining positioning in a core ontology for robotics.
651 In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ Interna-*
652 *tional Conference on. IEEE*; 2013, p. 1867–1872.

- [9] 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.
- [10] torga, M., Andreassen, 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}.
- [11] Huang, H.M., Messina, E., Albus, J.. Toward a generic model for autonomy levels for unmanned systems (alfus). Tech. Rep.; DTIC Document; 2003.
- [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.

- 685 [19] Modayil, J., Kuipers, B.. Autonomous development of a grounded ob-
 686 ject ontology by a learning robot. In: Proceedings of the national con-
 687 ference on Artificial intelligence; vol. 22. Menlo Park, CA; Cambridge,
 688 MA; London; AAAI Press; MIT Press; 1999; 2007, p. 1095.
- 689 [20] Suh, I.H., Lim, G.H., Hwang, W., Suh, H., Choi, J.H., Park, Y.T..
 690 Ontology-based multi-layered robot knowledge framework (omrkf) for
 691 robot intelligence. In: Intelligent Robots and Systems, 2007. IROS 2007.
 692 IEEE/RSJ International Conference on. IEEE; 2007, p. 429–436.
- 693 [21] Johnston, B., Yang, F., Mendoza, R., Chen, X., Williams, M.A..
 694 Ontology based object categorization for robots. In: Practical Aspects
 695 of Knowledge Management. Springer; 2008, p. 219–231.
- 696 [22] Lim, G.H., Suh, I.H., Suh, H.. Ontology-based unified robot
 697 knowledge for service robots in indoor environments. Systems, Man
 698 and Cybernetics, Part A: Systems and Humans, IEEE Transactions on
 699 2011;41(3):492–509.
- 700 [23] Stenmark, M., Malec, J.. Knowledge-based industrial robotics. In:
 701 SCAI. 2013, p. 265–274.
- 702 [24] Balakirsky, S., Kootbally, Z., Kramer, T.R., Pietromartire, A.,
 703 Schlenoff, C., Gupta, S.. Knowledge driven robotics for kitting ap-
 704 plications. Robotics and Autonomous Systems 2013;61(11):1205–1214.
- 705 [25] Gonçalves, P.. Towards an ontology for orthopaedic surgery, appli-
 706 cation to hip resurfacing. In: Proceedings of the Hamlyn Symposium
 707 on Medical Robotics. London, UK. ISBN 978-0-9563776-4-7; 2013, p.
 708 61–62.
- 709 [26] Haidegger, T., Barreto, M., Gonçalves, P., Habib, M.K., Raga-
 710 van, V., Li, H., et al. Applied ontologies and standards for ser-
 711 vice robots. Robotics and Autonomous Systems 2013;61(11):1215–1223.
 712 doi:\bibinfo{doi}{10.1016/j.robot.2013.05.008}. URL [http://dx.doi.](http://dx.doi.org/10.1016/j.robot.2013.05.008)
 713 [org/10.1016/j.robot.2013.05.008](http://dx.doi.org/10.1016/j.robot.2013.05.008).