Elsevier Editorial System(tm) for Robotics and Computer Integrated Manufacturing
Manuscript Draft

Manuscript Number: RCIM-D-14-00108R1

Title: Exploring the IEEE Ontology for Robotics and Automation for Heterogeneous Agent Interaction

Article Type: SI:Knowledge Driven Robotics

Keywords: Human-robot interaction; Robot-robot interaction; Ontologies for robotics and automation; Knowledge representation; Ontology engineering; Spatial information

Corresponding Author: Mr. Vitor Augusto Machado Jorge,

Corresponding Author's Institution: Universidade Federal do Rio Grande do Sul

First Author: Vitor Augusto Machado Jorge

Order of Authors: Vitor Augusto Machado Jorge; Vítor F Rey; Renan Maffei; Sandro Fiorini; Joel L Carbonera; Flora Branchi; João Pedro Meireles; Guilherme S Franco; Flávia Farina; Tatiana S da Silva; Mariana Kolberg; Mara Abel; Edson Prestes

Abstract: Spatial notions play a key role when humans and robots interact. Robotics & Automation (R&A) often involves diverse scenarios where heterogeneous robots must share their spatial knowledge to achieve a given goal. Such scenarios may become more complex when humans are also involved. This means humans and heterogeneous robots must share their spatial information about the world. For this purpose, the IEEE Ontologies for Robotics and Automation (ORA) Working Group started developing an ontology, called POS, with the purpose of defining the core notions required to share spatial concepts in the R&A domain. This paper evaluates the proposed ontology through a use case scenario involving both heterogeneous robots and human-robot interactions, showing how to define new spatial notions using POS. We discuss the experiment results presenting the ontology strengths as well as the future directions to be taken by the ORA group.

COVER LETTER

Experiment Background and Goals

The experiments pertaining this research were performed in the facilities of Universidade Federal do Rio Grande do Sul. The current work had three main purposes: specialize new spatial concepts using those available in the current version of the ORA standard; verify its strengths; and also possible limitations for the R\&A domain. The ORA standard was constructed considering the METHONTOLOGY methodology. This implies a continuous improvement and evaluation of the standard. This paper reflects this commitment.

The current work experimentally demonstrated the applicability of ORA in the context of R\&A. It also showed the benefits of using a common knowledge presentation strategy. The current experiment was also important to point future directions of the ORA group regarding future needs encompassing spatial notions.

Note: please use the main author to send us the updates on the paper, i.e.:
Vitor Augusto Machado Jorge
Universidade Federal do Rio Grande do Sul - Brazil
vitor_mj@yahoo.com

Highlights

The current work had three main purposes: specialize the spatial notions of the current version of the IEEE Ontologies for Robotics & Automation (ORA) standard; verify its strengths; and also possible limitations for the Robotics & Automation (R&A) domain. The ORA standard was constructed considering the METHONTOLOGY methodology. This implies a continuous improvement and evaluation of the standard. This paper reflects this commitment.

1. The current work experimentally demonstrated the applicability of ORA in the context of R&A.
2. It also showed the benefits of using a common knowledge presentation strategy.
3. The current experiment was also important to point future directions of the ORA group regarding future needs encompassing spatial notions.

***Detailed Response to Reviewers**

Dear reviewers,

We would like to thank you all for the input.

Summary:
We have modified the experiment section and clarified the difference between SUMO, CORA and our specialized notions. We have also made clear what were the points for improvement in the paper. Most of the Experiment section was modified to better explain what we actually do. We tried to address all the authors comments.

Reviewer 1: The generality of the case study can be seen as a strength due to large applicability of the approach. We have corrected the points you have raised related to figures and text (some of it was modified to improve readability). We believe, we have provided more information so that the scenario can be better understood. Regarding point 3, now we believe we have made clear the information exchanged and how the robots behave.

Reviewer 2: We were not clear enough in the previous version of the manuscript, so we changed the discussion to better address what we have detected in POS that must be improved. Regarding the quantitative results/analysis, an analysis of this sort would require a large group of people developing competing solutions using POS and other baseline approaches. At this time we do not have the resources to do it. It is also difficult to perform such achievement since this ontology is new and under development.

Reviewer 3: We tried to clarify the differences of spatial reasoning in the introduction and the related work. We have developed taxonomy diagrams differentiating concepts from SUMO, POS and our own. These along with text modifications may have solved the problem. We have cut some parts of the text and made some additional modifications to try to clarify the issues in the coordinate system definition section. Regarding the highlights file, we corrected the problem with acronyms.

Exploring the IEEE Ontology for Robotics and Automation for heterogeneous agent interaction

Vitor A. M. Jorge, Vitor F. Rey, Renan Maffei, Sandro Rama Fiorini, Joel L. Carbonera, Flora Branchi, João P. Meireles, Guilherme S. Franco, Flávia Farina, Tatiana S. da Silva, Mariana Kolberg, Mara Abel, Edson Prestes[1]

Bento Gonçalves 9500, Porto Alegre, Brazil

Abstract

Spatial notions play a key role when humans and robots interact. Robotics & Automation (R&A) often involves diverse scenarios where heterogeneous robots must share their spatial knowledge to achieve a given goal. Such scenarios may become more complex when humans are also involved. This means humans and heterogeneous robots must share their spatial information about the world. For this purpose, the IEEE Ontologies for Robotics and Automation (ORA) Working Group started developing an ontology, called POS, with the purpose of defining the core notions required to share spatial concepts in the R&A domain. This paper evaluates the proposed ontology through a use case scenario involving both heterogeneous robots and human-robot interactions, showing how to define new spatial notions using POS. We discuss the experiment results presenting the ontology strengths as well as the future directions to be taken by the ORA group.

Keywords: Human-robot interaction, Robot-robot interaction, Ontologies for robotics and automation, Knowledge representation, Ontology engineering, Spatial information

1. Introduction

Robotics and automation (R&A) systems play a key role in the current manufacturing landscape, since some of its main aspects, such as varying demands and dynamic environments, impose new challenges to the production. Even though an industrial robot must operate fast and precisely, these qualities are no longer enough. Machines are now expected to work together with humans and not in isolation from each other. Human-robot interaction (HRI) will not only be physical, but will also happen at the cognitive level. For the manufacturer it means better adaptation to different products and also a safer and more dynamic environment [1].

The domain of R&A must evolve to provide additional benefits to the whole society by participating more pervasively in everyday life. However, for reaching this goal, humans and robots should also effectively communicate and share their knowledge about the world among themselves. Nonetheless, current robots mostly depend on modeled environments and also generally require that the environment must adapt to them, not the opposite [2]. Unfortunately, the lack of capacity of robots to perform spatial reasoning is one of the main factors associated with these limitations. That is, a robot must understand the implications of given actions when objects and agents – other robots, people, or systems – are in specific positions/orientations.

This scenario only highlights the importance of *ontologies* for R&A systems. Ontologies provide a formal tool to describe the knowledge of a given domain. The result of the process of building an ontology is a knowledge artifact, which formally describes the main concepts, relations, and axioms in a given domain [3]. The relevance of ontologies for R&A systems is broad. They provide a vocabulary that is formally defined, which help to ensure a common understanding among various stakeholders involved in the domain. Ontologies can also be implemented as semantically-rich data models, which enable efficient and reliable data integration and information exchange between different agents

[1] Universidade Federal do Rio Grande do Sul

or systems. Finally, ontology tools offer reasoning tools, such as consistency checking and taxonomic reasoning, which can be leveraged by robots in tasks such as planning and navigation.

Spatial reasoning is fundamentally entangled with concepts such as position, orientation, and pose. Bateman and Farrar [4] discuss the ontological nature of space and the need of a suitable ontological account for space-related concepts to improve the semantic understandings among different robotic systems. Even though most spatial concepts can be seen as trivial, they have been subject of debates and controversies [5], resulting in different conceptions. Knowledge representation and qualitative reasoning communities identified different notions – such as topology, orientation, shape, size, distance, positioning, etc – which are important for spatial reasoning [6]. However, there is a lack of approaches that allows the representation and reasoning considering these notions in an integrated way, with a unique perspective of them. The proposed approaches usually handle few of these notions, without expliciting the underlying ontological commitments.

The IEEE Ontology for Robotics & Automation (ORA) [7] [8] is focused on R&A knowledge standardization. One of its underlying proposed ontologies is POS [3], encompassing concepts associated with positioning. In this work we extend spatial concepts provided by POS to solve problems in the R&A domain. The use of the ORA standard is illustrated and discussed in an interaction application of positioning concepts in a robot-robot and human-robot interaction.

This paper is divided in the following sections. Section 2 briefly describes the Core Ontology for Robotics and Automation. Section 3 details the spatial concepts available in the standard. Section 4 reviews approaches from other authors regarding spatial knowledge representation. The experiment is detailed in Section 5. Finally, Section 6 discusses the standard strengths and points for improvement.

2. Core Ontology for Robotics and Automation

The *IEEE RAS Ontology for Robotics and Automation Working Group* (ORA WG) [7] aims to standardize knowledge representation in the robotics and automation (R&A) domain. The idea is to to develop a set of ontologies to describe the major sub-domains within R&A, such as industrial and service robotics. The ORA WG comprises different sub-groups. One of them is the Upper Ontology/Methodology (UpOM) group, focused on the development of a Core Ontology for R&A (CORA), which specifies the main concepts and relations of the field. Its main goal is to serve as an integration tool for the different subontologies within and possibly outside the group.

CORA was developed using the Suggested Upper Merged Ontology (SUMO) [9] [10]. Its main concepts were introduced by different authors [7] [8]. The standard is currently divided into CORA and other three supporting ontologies:

- CORAX – defines the required general concepts not specific to the R&A domain (design, physical environment, interaction, etc.), but not covered in SUMO;
- CORA – defines the core R&A concepts such as robot, robotic system, etc;
- RPART – defines the most general kinds of devices that can constitute a robot, extending those already present in SUMO;
- POS – defines those concepts associated with spatial knowledge (position and orientation represented as points, regions and coordinate systems)

This paper uses the ORA standard ontology, specially POS, which is described in the next section.

3. POS: Positioning Ontology

Carbonera et al. [3] introduced the core concepts associated with positioning in the Positioning Ontology (POS) – part of the ORA standard. Below we briefly describe it, presenting its taxonomy (see Figure 1) along with its core concepts – *position* and *coordinate systems*.

Position is defined as a physical quantity in SUMO. POS separates positions in *points* (PositionPoint) or *regions* (PositionRegion), which is defined in relation to a given *position coordinate system*. Besides that, POS states that every position coordinate system must be *grounded* in a *reference physical object*. *Orientation* is defined in a similar way. That is, orientation is either a point[2] or a

[2]During the writing of this manuscript, ORA changed the

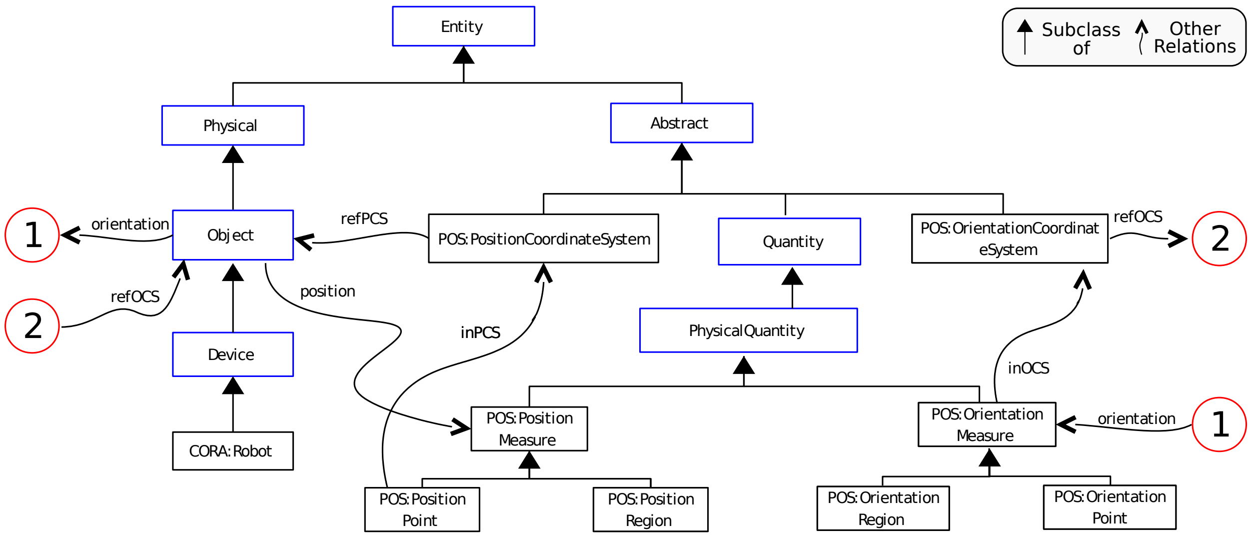


Figure 1: POS ontology overview. Blue boxes show the concepts from SUMO, while black boxes present the concepts from POS or CORA.

region of an object in a given *orientation coordinate system*, which is also grounded to a reference object.

Coordinate systems are abstract SUMO entities. Usually, an agent arbitrarily chooses an appropriate coordinate system as the global reference frame – the *global coordinate system* (GCS). *Local coordinate systems* (LCS) are defined in relation to GCS by hierarchical links. This notion of hierarchy is also arbitrarily defined according to the needs of the agent.

POS defines *transformation* – for both orientation and position points – as a transitive function that transforms points in a coordinate system to points in another coordinate system. There is a mapping from one coordinate system to another if and only if there is a transformation that maps all points in one coordinate system to another. As a matter of fact, when two coordinate systems share a direct or indirect connection in the hierarchy, there is a mandatory transformation between them.

At the core of POS is the general notion of *spatial operator* (see the taxonomy in Figure 2). Spatial operators act as generators of orientation and position regions over a reference object. For instance, we can apply operators to a robot to generate position regions constituting its back, front, left and right regions.

It is also possible to define the *pose* of an object as a combination of position and orientation information. The information required for specific applications can be especialized from these core concepts in a straightforward way. In this paper, we put the POS ontology through a test where it is used by heterogeneous robots and human agents to perform a task in the R&A domain.

4. Related Work

The literature provides different approaches for representing spatial knowledge, since they are important in many different domains.

Freeman [11] discusses how people encode spatial relations and their relationship with English names. He makes use of mathematical formalisms to model the semantic information of such terms. The author studies the major psychological aspects associated to human picture encoding, associating them to spatial relation encoding. He is able to model binary and ternary relations, such as *between* and *right*. Kuipers [12] models spatial relations targeting routes, their underlying topological structures, and the actions required to follow such route. He also defines a position in relation to a coordinate system and regions (as groups of places). Unfortunately, their framework is not generic enough to encompass the R&A domain. In particular, inference

name of orientation points to orientation values since orientation can be seen as a vector, which led to confusion. We have chosen to keep the original name used in our experiments.

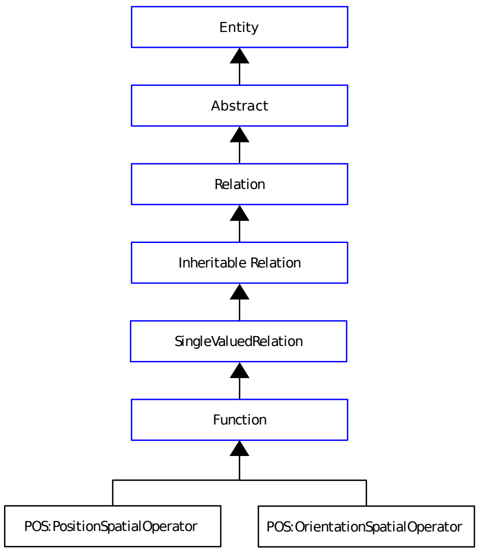


Figure 2: Operator relation. Blue boxes show the concepts from SUMO, while black boxes present the concepts from POS.

rules are 2D, which, for instance, is not enough for industrial or Geographic Information Systems applications. Hudelot et al. [13] propose an ontology of spatial relations focused on helping medical image interpretation, recognition of image structures and their spatial arrangements. They also model fuzzy representations of spatial concepts aiming to mitigate the existing semantic gap between knowledge representation and perception of spatial relations and their intrinsically vague linguistic representations.

Bateman and Farrar [4] propose a unified ontological framework for representing qualitative (relative) positioning in space, but they do not provide explicit treatment of some quantitative positioning notions, like position of an object according to a coordinate system. On the other hand, other approaches, such as Ye et al. [14], represent positions, coordinate systems, and relative positions, although they do not make clear statements about their ontological commitments. For instance, they do not provide a clear formal description of what is a coordinate system.

SUMO [9] [10] defines the concept of region. Regions are topographic locations in space – be it real or imaginary. While it defines a broad range of spatial relations that includes mereological and topological notions, it lacks the concept of coordinate system and exact, quantitative position in it.

5. Use case: Human-robot and robot-robot interaction in a manufacturing task

In order to evaluate POS, we developed an experiment where two robots collaborate to deliver an object to a human. Our goal is to illustrate the possibility of cooperation among heterogeneous robots and humans using CORA by specializing its generic concepts. This scenario can be extrapolated to more complex manufacturing systems, such as assembly lines, in a straightforward way.

The specific agents participating in this task are:

- **human** – a human agent using a custom user interface made in python;
- **manipulator** – an Aldebaran NAO H25 humanoid robot controlled by a python application;
- **transporter** – a Pioneer 3DX mobile robot, equipped with a SICK LMS-200 laser rangefinder, controlled by a c++ application.

The object to be delivered, named **cargo**, is a simple pen, which is originally in the possession of the **manipulator** and must end in the possession of the **human**.

Individually, each agent has limitations. The **human** needs the pen – i.e. the **cargo**. Unfortunately, he is busy performing a higher priority task, which requires his uninterrupted attention. This can be seen as a mobility constraint. However, he can easily identify nearby objects and interact with them when necessary, for example, to receive the **cargo**. The **manipulator** is a humanoid robot endowed with the ability to grab, lift and drop the **cargo**. It has a good visual recognition system, capable of detecting objects with reasonable accuracy. Nevertheless, its locomotion is limited due to low battery autonomy and speed. Finally, the **transporter** can move across large distances avoiding obstacles and carrying considerable payloads. However, it has no way to manipulate objects. That is, the **cargo** must be placed or removed from it by some other agent. The **transporter** has also limited object recognition capabilities, due to its sensor limitations. Thus, it must rely on the **manipulator** or the **human** to tell it who is next to it.

As we have seen by their limitations, in order to complete the task, the agents must share positioning information, e.g. position of the cargo, relative position and orientation of nearby objects, etc.

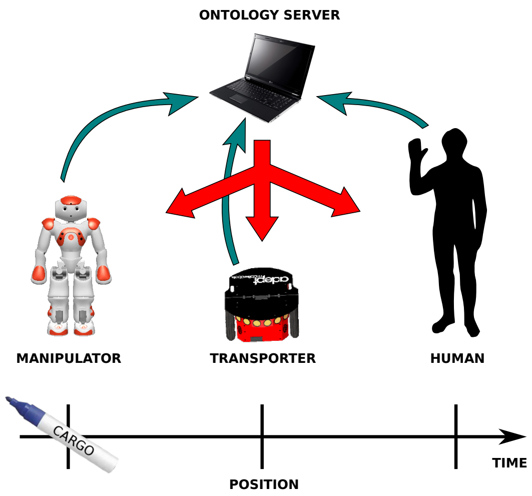


Figure 3: Experiment overview.

This can be accomplished using ontologies, in this case, ORA. For this experiment, we chose to represent the existing ORA ontologies using the Web ontology language (OWL) [15] – a W3C recommendation for semantic web applications based on description logics. The ontology server runs a Java application which loads the OWL ontologies representing CORA. Once the agents are connected to the ontology server, they receive all the server's ontology information and send their own. This process continues throughout the whole experiment with the agents sending the knowledge they acquire to the server and receiving new knowledge that the server inferred or received from other agents. This means that all agents share the same knowledge base and communicate among themselves through the ontology server. The key advantage of ontology-based communication is that the agents share the same spatial concepts and are able to communicate those using the same shared vocabulary with explicit semantics.

Figure 3 shows an overview of the experiment. The three agents individually send their information to the ontology server, as shown by the green arrows. The broadcasting of this information by the server is shown by the red arrows. The agents collaborate to achieve the goal, which is to take the **cargo** from the **manipulator** to the **human**.

5.1. Definition of the coordinate systems

When using POS, one needs to define the position and the orientation coordinate systems to be used.

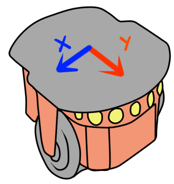


Figure 4: Definition of the agent orientation.

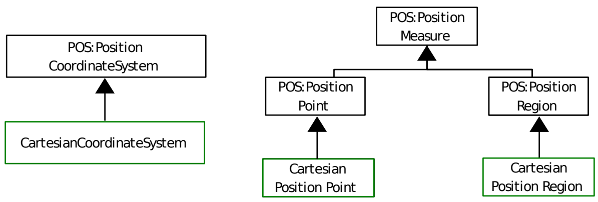


Figure 5: Concepts associated with the position coordinate system. The concepts in black are defined in POS, while those in green were created for the experiment.

In this experiment, the required qualitative spatial operations can be defined in a 2-dimensional coordinate system, so that every point is represented as a 2-tuple

$$\eta(p) = (x(p), y(p))$$

where $(x(p), y(p))$ represents, respectively, x and y coordinates of the position point in the Cartesian canonical orthonormal base with standard orientation.

Each coordinate system will have an agent as its reference object, so that the point $(0, 0)$ represents the agent's quantitative position in it. The y axis will point the current orientation of the agent and the x axis will point to the right, as seen in Figure 4.

We will name this type of coordinate system a *CartesianCoordinateSystem*, or Ccs for short. Points in it will be *CartesianPositionPoints* and regions in it *CartesianPositionRegions*. Figure 5 highlights the taxonomy of the concepts associated with the position coordinate system.

Since POS separates orientation from position, the orientation of the agent will have its own coordinate system. We will call it *AngularOrientationCoordinateSystem* or AngularCS for short. Figure 6 highlights the taxonomy of the concepts associated with the orientation coordinate system.

To attribute an *AngularOrientationPoint* with value o_v to some object o_1 in relation to o_2 means that the Ccs base of o_2 can become the base of o_1

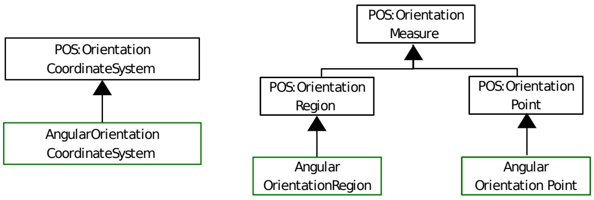


Figure 6: Concepts associated with the orientation coordinate system. The concepts in black are defined in POS, while those in green were created for the experiment.

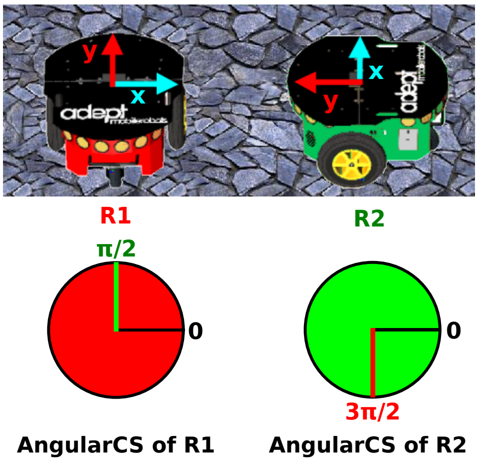


Figure 7: Orientation of R2 as seen by R1 and orientation of R1 as seen by R2.

when rotated by o_v degrees. For example, the coordinate system of robot R1, in Figure 7, can become the coordinate system of R2 by applying a rotation of $\pi/2$. This means that, in R1's AngularCS, the orientation of R2 is represented by a point of value $\pi/2$. Analogously, R1's orientation point has value $3\pi/2$ in R2's AngularCS.

5.2. Spatial operators

Spatial operators in POS are functions that, when applied to objects (e.g. robot, person, tool), generate regions. They can be used to express qualitative spatial relations between objects. In our use case, we devised spatial operators regarding four aspects – visibility, proximity, placement and relative orientation – pertinent for the task completion. The first three aspects are associated to position operators, while the last one is associated to orientation operators. The taxonomy of the spatial operators defined for the use case is presented in Figure 8.

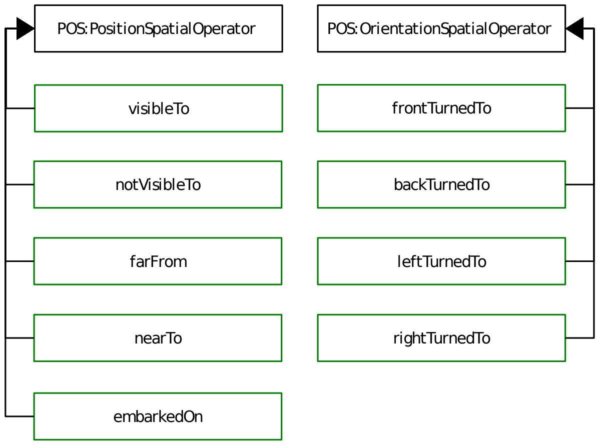


Figure 8: Taxonomy of the spatial operators. The concepts in black are defined in POS, while those in green were created for the experiment.

5.2.1. Position operators

When a position operator is applied to a given object it generates a position region. In this paper, expressions using position operators are represented as follows:

```
Position OT OP OR
```

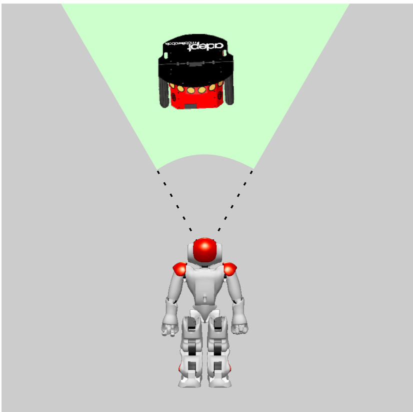
where OP is an operator valid for objects OR and OT, when OT is positioned at the region generated by applying OP over OR.

For example, regarding object visibility, the operator `visibleTo`, applied to a robot, generates the position region constituting the space that is visible to that robot. Analogously, the operator `notVisibleTo` represents the region of space that cannot be seen by the robot. In Figure 9, we have that **transporter** is positioned at the region generated by the operator `visibleTo` applied over the **manipulator**. That is,

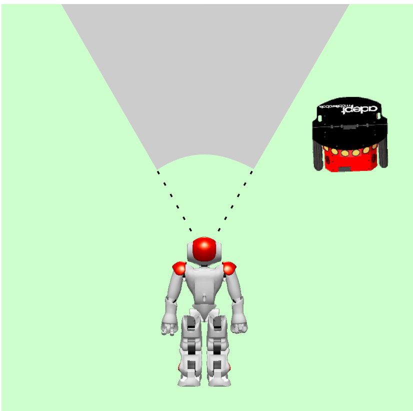
```
Position Transporter visibleTo Manipulator.
```

The `visibleTo` operator is used by the **human** and the **manipulator**. The latter has a visual recognition API able to recognize objects based on their features. The robot was previously trained with images of the **transporter** in order to be able to recognize it.

The second type of position operator defined in this work is the one associated to placement. To represent that an agent is carrying some object, we have defined the operator `embarkedOn` which,

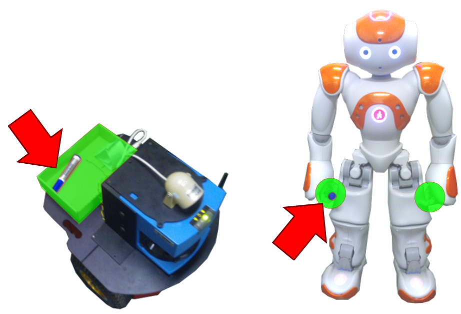


(a) **transporter** visible



(b) **transporter** not visible

Figure 9: Example of `visibleTo` operator. Objects in the green area of image (a) are visible to **manipulator**, while objects in the green area of image (b) are not.



(a) (b)

Figure 10: Example of `embarkedOn` operator. Images (a) and (b) depict the **transporter** and the **manipulator** used in the experiment, highlighting the position regions representing where the **cargo** can be stored – generated by the `embarkedOn` operator applied over each robot.

when applied to an agent, generates the region corresponding to its shipment compartment. Figure 10 shows these regions for **transporter** – the compartment at its back – and **manipulator** – its hands.

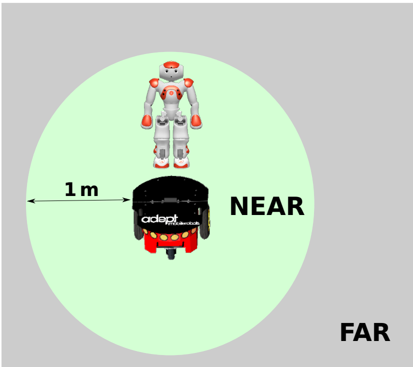
The third type of position operator is associated to proximity. In our case study, the **transporter** must perceive and avoid nearby objects during navigation. We represent such qualitative notions using operators `nearTo` and `farFrom` that, when applied to some object, generate regions considered as near and far, respectively. The **transporter** uses its laser range-finder to detect objects in its front. This detection is done by searching for discontinuities in adjacent laser readings. When the approximate position of an object is less than $1m$, it is considered to be in the region generated by the `nearTo` operator. Figure 11 shows these regions for **transporter**.

5.2.2. Orientation operators

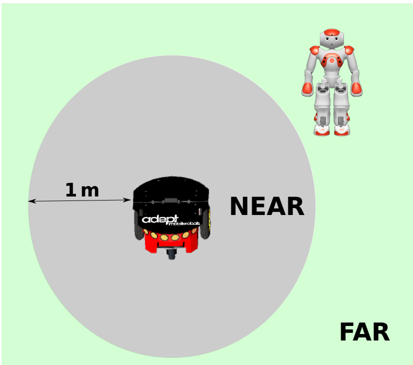
The orientation of a robot is another important spatial information. For example, our task requires two robots to be aligned, therefore orientation operators are used to determine if a robot is facing the desired direction. The representation of validity for orientation operator is analogous to the one presented for position. That is,

```
Orientation OT OP OR
```

For instance, `"Orientation Transporter leftTurnedTo Manipulator"`, represents a valid



(a) **manipulator** near to **transporter**



(b) **manipulator** far from **transporter**

Figure 11: Example of `nearTo` and `farFrom` operators. Objects in the green area of image (a) are near to **transporter**, while objects in the green area of image (b) are far from **transporter**.

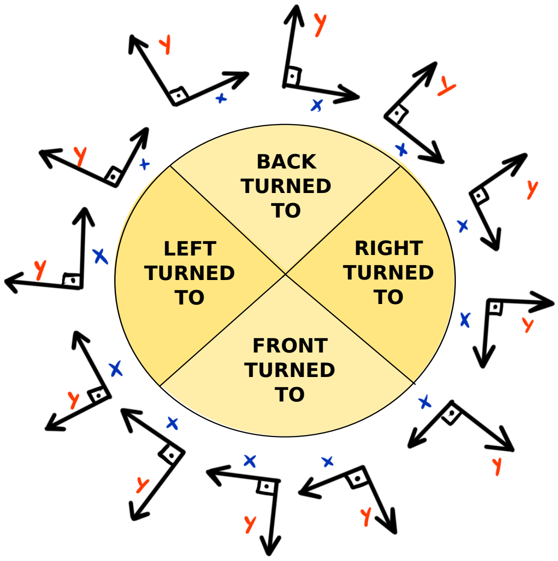


Figure 12: Qualitative regions defined for the orientation coordinate system.

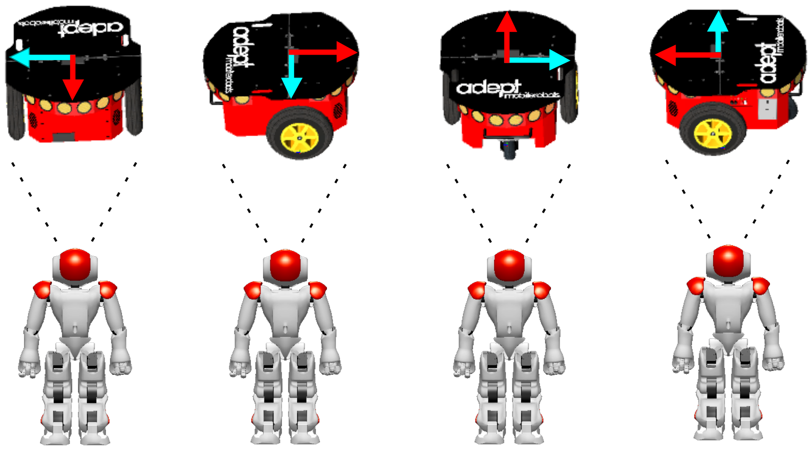
orientation for the **transporter** according to **manipulator**. That is, **transporter**'s left is turned to **manipulator**.

Using image recognition techniques, the **manipulator** can determine the orientation of objects in the environment, but this information is not precise. It can only obtain a region representing a range of orientations, as seen in Figure 12. The **manipulator** detects the orientation of the **transporter** using the orientation operators `leftTurnedTo`, `rightTurnedTo`, `backTurnedTo` and `frontTurnedTo`, that are extracted from the vision recognition API. Figure 13 shows examples of possible orientation values.

5.3. Implementing a Cargo Transportation Strategy

We developed a collaborative strategy to perform the cargo transportation that is essentially composed of two acts. First, the transporter must approach the manipulator to receive the cargo, and second, must approach the human to deliver the cargo. In both cases, task completion is conditioned to successful spatial knowledge transfer among agents.

Table 1 summarizes in six steps the complete strategy, along with the spatial information perceived by the agents. Agents always transmit the information to the ontology server, which is used to



(a) (b) (c) (d)

Figure 13: Example of qualitative orientation operators. (a) Transporter frontTurnedTo Manipulator. (b) Transporter rightTurnedTo Manipulator. (c) Transporter backTurnedTo Manipulator. (d) Transporter leftTurnedTo Manipulator.

#	Activity	Agent	Message
1	**human** requests **cargo**	**human**	`Position Cargo notVisibleTo Human`
2	**manipulator** informs **cargo** position	**manipulator**	`Position Cargo embarkedOn Manipulator`
3	*transporterApproaches(**manipulator**)*	—	—
4	**manipulator** delivers **cargo**	**manipulator**	`Position Cargo embarkedOn Transporter`
5	*transporterApproaches(**human**)*	—	—
6	**human** collects **cargo**	**human**	`Position Cargo embarkedOn Human`

Table 1: Strategy

update the spatial knowledge database. The new information is then broadcasted to all agents.

In the start of the experiment, the **human** does not see the **cargo**, so he transmits this information using a custom python interface running in a mobile phone (step 1). The processed information is received by the **transporter** and the **manipulator**. This information triggers the collaboration initiative. Since the **transporter** cannot detect the **cargo** by itself, it awaits for **cargo** position information. On the other hand, the **manipulator** knows the **cargo** position – conveniently, in its hands – thus, it informs that the **cargo** is embarked on its shipment compartment (step 2). Then, the **transporter** searches and moves toward the **manipulator** in order to receive the **cargo** (step 3). This step involves exploring the environment, using spatial information from both agents, to find the **manipulator**. The procedure associated with this step – *transporterApproaches* – is detailed later in this section. When the **transporter** is aligned with the **manipulator**, the latter prepares itself to deliver the **cargo**. That is, the **manipulator** slowly turns itself to get a proper alignment between its arm and the **transporter**'s shipment compartment. This is done with the aid of the **manipulator**'s visual recognition system. Next, the cargo is dropped and the **manipulator** informs the change in **cargo**'s position (step 4). In possession of the **cargo**, the **transporter** goes to the **human** in order to complete the experiment, using the same procedure of step 3. Finally, after the alignment between **human** and **transporter**, the former collects the **cargo** and informs the new **cargo**'s position, ending the experiment (step 6).

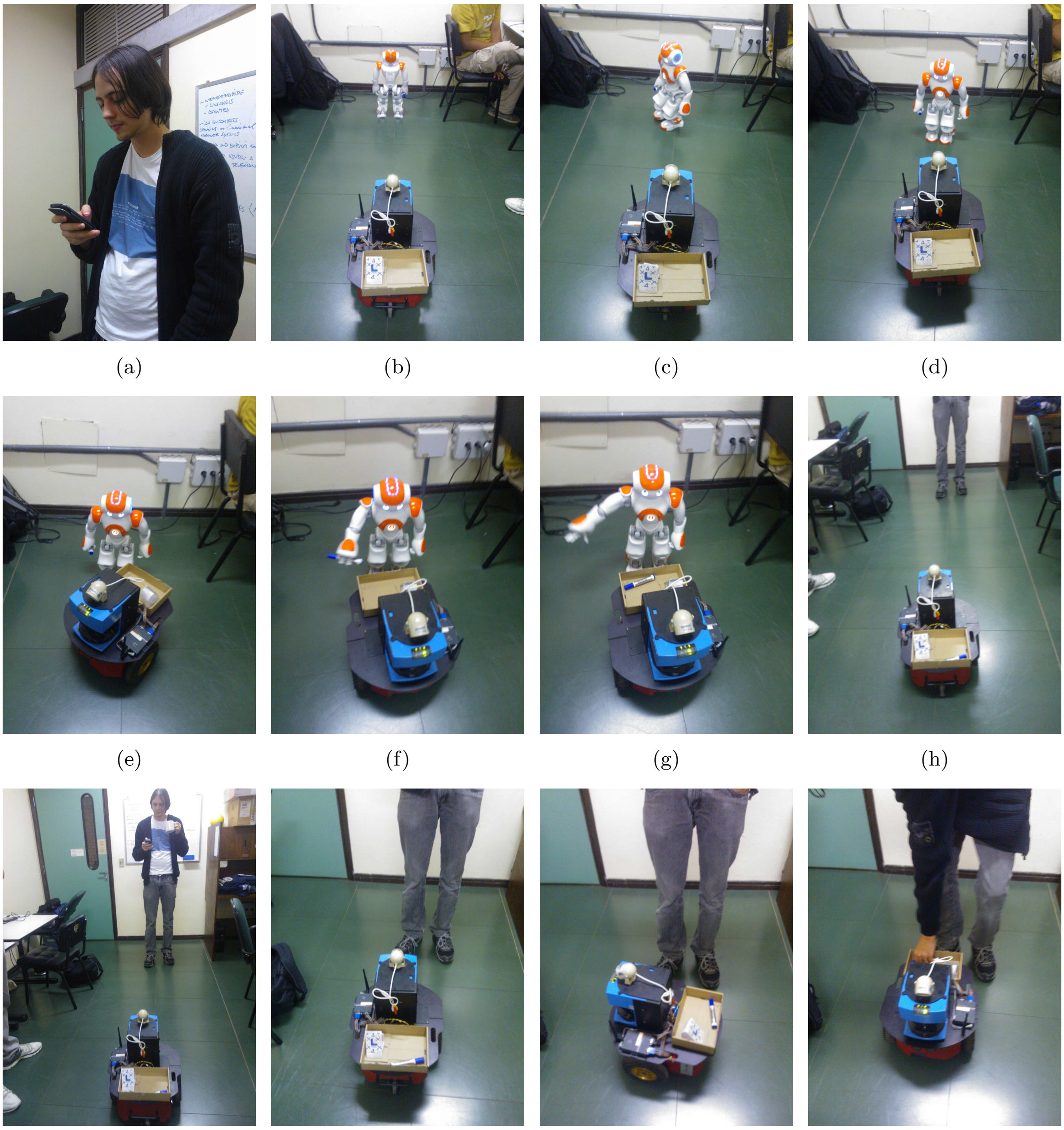
Table 2 describes the procedure *transporterApproaches* performed in steps 3 and 5 of Table 1, with the **goalAgent** corresponding to the **manipulator** in the first case, and to the **human** in the second. The purpose of this procedure is to make the **transporter** approach the **goalAgent** and prepare itself for the **cargo**'s transfer. In order to approach the **goalAgent**, the **transporter** must first find it. This is performed by detecting salient objects using its laser range-finder (step 1). Due to sensor limitations, the **transporter** does not know if the detected object corresponds to the **goalAgent**. Thus, it must approach the candidate to confirm its identity (step 2). During this phase, the transporter performs obstacle avoidance and trajectory correction to end up facing the candidate. When the **transporter** gets near the object (around $1m$), it informs that the object is nearby. Then, to confirm that the candidate is in fact the **goalAgent**, the latter turns 360° searching its surroundings for the **transporter**, using its visual recognition system (step 3). The confirmation will occur if the two agents are aligned, i.e. the front of **transporter** is facing the front of **goalAgent** (step 4a). Otherwise, the candidate is not the **goalAgent** (step 4b). Therefore, the **transporter** tries to find another candidate, what leads back to step 1. Additionally, during the search, if the **goalAgent** detects the **transporter** in an orientation different from the desired one, the **transporter** can use this orientation to correct itself, e.g. **transporter** will turn left if **goalAgent** observes the **transporter**'s left. In the end, after successful alignment, the **transporter** gets close to **goalAgent** (around $20cm$) and turns itself until its shipment compartment is within the reach of the **goalAgent** (step 5).

The proposed robotic system was able to achieve its function, that is, it could provide the **human** with the **cargo**. A video of a run of the experiment is available at `http://www.inf.ufrgs.br/phi-group/`.

Figure 14 shows an ordered sequence of pictures from the experiment. In (a), the **human** informs the ontology server that he's not seeing the **cargo**. In (b), the **transporter** starts to approach the **manipulator**, after receiving the information that the **cargo** is with the **manipulator**. When the **transporter** gets near the **manipulator**, the latter starts to turn in search for the **transporter**, as shown in (c). We can see in (d) that, after the **manipulator** informed that the **transporter** is in front of it, the **transporter** gets close to the **manipulator**. In (e), the **transporter** turns its back to the **manipulator**, in order to receive the **cargo**. Both (f) and (g) show the **manipulator** delivering the **cargo** to the **transporter**. From (h) to (l), we can see the same process being applied to the **human**, the difference being that the **human** collects the **cargo** in the end of the process.

6. Discussion

In this paper, we experimentally demonstrated the applicability of ORA standard in a common task that required an heterogeneous groups of robots to communicate and coordinate. CORA and related ontologies helped in guaranteeing a common understanding among all agents involved in the task. As showed in Section 5, ORA allowed the



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 14: Pictures of the real experiment.

#	Activity	Agent	Message
1	Search for **goalAgent**	**transporter**	—
2	Approach candidate	**transporter**	`Position Candidate1 nearTo Transporter`
3	Try to align agents	**goalAgent**	—
4a	If it succeeds	**goalAgent**	`Orientation Transporter frontTurnedTo GoalAgent`
4b	else, go to step 1	**goalAgent**	`conditional to scenario`
5	Prepare for **cargo** deliver	**transporter**	`Orientation Transporter backTurnedTo GoalAgent`

Table 2: procedure *transporterApproaches(GoalAgent)*

cooperation among heterogeneous robots and humans, i.e., two robots successfully worked together and with a human agent to provide him a missing tool. The development process spanned three different languages and a reasonably large group of developers working concurrently. This was mainly due to the standardized communication provided by POS. As a result, developers and domain experts could better understand what were the spatial concepts involved in the experiment. This included the understanding of the available spatial information for each system object as well as those that the robotic system had to achieve. This illustrates how ORA can be used for solving tasks in R&A domain that require a shared conceptualization among the agents (robots and humans). There are many scenarios that pose this requirement and that could benefit from the use of ORA standard, including supply-chain operations, assembly lines, medical applications, service robotics for the home, ubiquitous robotics, and many others.

Our experimental evaluation also substantiates the claim that ORA standard provides a general basis for the formal specification of spatial primitives. For instance, POS allows for the representation of n-dimensional coordinate systems, essential for the R&A domain due to its vast use in different areas.

Nevertheless, all spatial information in the presented experiment represents the *current* state of the system, such as those used by any nonlinear estimation system [16]. This might not be enough for other domains where it is necessary to keep track of the evolution of spatial information in time, i.e. the history of all the agents positions. For this case, we are extending ORA with a *diachronic* version of POS.

Finally, we are still in the process of extending the ontology to encompass a larger portion of the R&A domain. For instance, at first we considered implementing a scenario where drones rely on geographic information systems (GIS) to navigate around the environment. This example has natural requirements for spatial information about positions, mappings and coordinate system transformations. Notwithstanding, there are a few requirements that were not met by POS at the time, such as region to region transformations and n-ary spatial operators, e.g. between. The problem with n-ary operators in POS is that operators are applied over a single object, resulting in regions that are only present at its coordinate system. These issues are currently being addressed by the ORA Working Group.

References

[1] A. Bicchi, M. A. Peshkin, J. E. Colgate, Safety for physical human-robot interaction, in: B. Siciliano, O. Khatib (Eds.), Springer Handbook of Robotics, Springer Berlin Heidelberg, 2008, pp. 1335–1348. `doi:10.1007/978-3-540-30301-5_58`. URL `http://dx.doi.org/10.1007/978-3-540-30301-5_58`

[2] C. Heyer, Human-robot interaction and future industrial robotics applications, in: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, 2010, pp. 4749–4754. `doi:10.1109/IROS.2010.5651294`.

[3] J. Carbonera, S. Rama Fiorini, E. Prestes, V. Jorge, M. Abel, R. Madhavan, A. Locoro, P. Goncalves, T. Haidegger, M. Barreto, C. Schlenoff, Defining positioning in a core ontology for robotics, in: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, 2013, pp. 1867–1872. `doi:10.1109/IROS.2013.6696603`.

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

[5] M. Jammer, Concepts of Space: The History of Theories of Space in Physics: Third, Courier Dover Publications, 2013.

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

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

[8] E. Prestes, J. L. Carbonera, S. R. Fiorini, V. A. M. Jorge, M. Abel, R. Madhavan, A. Locoro, P. Goncalves, M. E. Barreto, M. Habib, A. Chibani, S. Gerard, Y. Amirat, C. Schlenoff, Towards a core ontology for robotics and automation, Robotics and Autonomous Systems 61 (11) (2013) 1193 – 1204, ubiquitous Robotics. doi:http://dx.doi.org/10.1016/j.robot.2013.04.005.
URL http://www.sciencedirect.com/science/article/pii/S0921889013000596

[9] I. Niles, A. Pease, Origins of the ieee standard upper ontology, in: Working Notes of the IJCAI-2001 Workshop on the IEEE Standard Upper Ontology, 2001, pp. 4–10.

[10] I. Niles, A. Pease, Towards a standard upper ontology, in: Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001, FOIS '01, ACM, New York, NY, USA, 2001, p. 29. doi:10.1145/505168.505170.
URL http://doi.acm.org/10.1145/505168.505170

[11] J. Freeman, The modelling of spatial relations, Computer Graphics and Image Processing 4 (2) (1975) 156 – 171. doi:http://dx.doi.org/10.1016/S0146-664X(75)80007-4.
URL http://www.sciencedirect.com/science/article/pii/S0146664X75800074

[12] B. Kuipers, Modeling spatial knowledge*, Cognitive Science 2 (2) (1978) 129–153. doi:10.1207/s15516709cog0202_3.
URL http://dx.doi.org/10.1207/s15516709cog0202_3

[13] C. Hudelot, J. Atif, I. Bloch, Fuzzy spatial relation ontology for image interpretation, Fuzzy Sets and Systems 159 (15) (2008) 1929 – 1951, from Knowledge Representation to Information Processing and Management Selected papers from the French Fuzzy Days (LFA 2006). doi:http://dx.doi.org/10.1016/j.fss.2008.02.011.
URL http://www.sciencedirect.com/science/article/pii/S0165011408001012

[14] J. Ye, L. Coyle, S. Dobson, P. Nixon, A unified semantics space model, in: J. Hightower, B. Schiele, T. Strang (Eds.), Location- and Context-Awareness, Vol. 4718 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 103–120. doi:10.1007/978-3-540-75160-1_7.
URL http://dx.doi.org/10.1007/978-3-540-75160-1_7

[15] D. L. McGuinness, F. van Harmelen, Owl web ontology language overview, Tech. Rep. REC-owl-features-20040210, W3C (2004).

[16] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series), Intelligent robotics and autonomous agents, MIT Press, Cambridge, MA, USA, 2005.
URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0262201623

LaTeX Source Files

Click here to download LaTeX Source Files: Latex Files.zip