Knowledge Representation and Reasoning for Perceptual Anchoring

Jonas Melchert and Silvia Coradeschi and Amy Loutfi
AASS Mobile Robotics Lab
Örebro University
S-70182 Örebro, Sweden
Email: *firstname.lastname*@aass.oru.se

Abstract

In this work we report results on the use of symbolic knowledge representation and reasoning (KRR) for perceptual anchoring. This is the creation and maintenance of a connection between symbolic and perceptual description that refer to the same physical object in the environment. We extend an anchoring framework to manage the symbolic information in a KRR system, and to exploit this knowledge and the inference mechanism to recover from failures in the anchoring of symbols. We show a simulated scenario where the system communicates with a user to interactively resolve an ambiguous description using the knowledge base, and in particular spatial relations. This is a first step towards a KRR-supported anchoring framework, that we will use for human-robot communication.

1. Introduction

Robotic and intelligent systems intending to interact at a symbolic level, require a connection between the abstract and the physical levels of representation. The term *anchoring* has been used in robotics to refer the creation of such symbol-percept connections that correspond to the same physical objects. To date, anchoring systems have proven useful for systems using planners and multi-modal sensing modalities. Most of these applications have focused on the autonomous robot with little interaction with human users, resulting in arbitrary labels for symbols.

A recent trend in robotics is moving towards the notion of *symbiotic robotic systems* which consist of a robot, human and (smart) environment cooperating together in performing different tasks [7]. In such systems a richer symbolic representation is required both to facilitate the interaction with humans and to store properties about objects that can come from different sensing devices. In this work we connect a knowledge representation and reasoning (KRR) system to an anchoring module which is able to provide extended reasoning about anchored objects. Using the tools of the anchoring module and the knowledge base a deeper human-robot or robot-robot interaction can be obtained.

To illustrate the benefit of the knowledge representation to the system, we compare results from a previous work [15] which considered how spatial relations are used to improve anchoring. In this previous work, spatial relations were coded and stored inside the anchoring module and could be used to locate objects and to disambiguate insufficient object descriptions. In contrast, the KRR system allows us to model spatial relations on a conceptual level, and the information is now managed inside the knowledge base. As a result, queries, including spatial relations, can be more advanced and are executed on the knowledge base.

This paper is organised as follows: Section 2 describes the computational framework used for anchoring and some aspects of the anchoring problem. In Section 3 we introduce the knowledge representation system and its coupling with the anchoring process. Section 4 describes the implementation of the system with spatial relations and an example shows it's functioning. Section 5 discusses our approach with respect to related work and mentions future work.

2. Perceptual Anchoring

As stated in the introduction, anchoring is "the process of creating and maintaining the correspondence between symbols and sensor data that refer to the same physical objects" [6].

In our work we use the computational framework for anchoring introduced in [5]. Following that, the main parts of anchoring are: a *symbol system*, including a set of individual symbols (identifiers) and a set of predicate symbols; a *perceptual system*, providing a set of percepts (a percept is a structured collection of measurements assumed to originate from the same physical object); and a *predicate grounding relation* which encodes the correspondence between the predicate symbols and compatible values of measurable attributes. The symbol-percept correspondence for a specific object is reified in a data structure $\alpha(t)$, called an *anchor*. It is indexed by time as the created links are dynamic and evolve over time. At each time instance t, $\alpha(t)$ contains a symbol identifying the object and a percept generated by the latest observation of the object. The *symbolic description* of an anchor is the set of predicates describing the symbolic properties of an object, and the *perceptual signature* of an anchor is the set of perceptual attribute values of an object.

Creation of anchors can occur both in a top-down and a bottom-up fashion: bottom-up acquisition is triggered when percepts from the sensory system can not be associated to existing anchors, and top-down acquisition occurs when a given symbol with a symbolic description needs to be anchored to a perceptual signature (such requests may come from a top-level task planner). Once an anchor has been created it is continuously tracked and its perceptual signature updated. These functionalities are realised through respective routines in the framework.

For the case of top-down anchoring, two important features have to be highlighted [6]:

Complete vs. Partial Matchings: Matchings between a symbolic description and a perceptual signature can be *partial* or *complete*. Given a percept π and a description σ, we say that π *fully matches* σ if each attribute in π matches a property in σ and vice-versa; π *partially matches* σ if each attribute in π matches a property in σ; otherwise π does not match σ.

Definite vs. Indefinite Descriptions: The given symbolic descriptions can be either *definite* or *indefinite*: a description is definite if it denotes a unique object, like for example "my coffee-cup on the table"; an indefinite description does not require that the object is unique, but that the object corresponds to the description, like for example "a coffee-cup".

Given a symbolic description and information about whether this is a definite or indefinite description, the anchoring module has to find (a) matching candidate(s) and has to detect possible ambiguities that can arise. The following table, from [4], summarises the cases that can occur:

	#Matches		Result	
Case	full	partial	definite	indefinite
1	0	0	Fail	Fail
2	0	1+	Fail	Fail
3	1	0	OK	OK
4	1	1+	OK/Fail	OK
5	2+	any	Conflict	OK

If an unambiguous match is found the anchoring module selects that candidate for anchoring. In case of a failure the anchoring process can make use of a task planner to create and execute a recovery plan with the aim of searching for unperceived objects, or collecting more perceptual properties that would allow a disambiguation, as discussed in [4].

In case 4, the presence of several candidates for a definite description can constitute a problem: although the complete candidate can be selected for anchoring, a more careful approach would try to rule out the partially matching candidates by creating and executing a recovery plan to observe their missing properties.

Case 5 for a definite description constitutes a serious problem: the situation cannot be resolved by active perception to observe more properties, but the description is ambiguous and can be considered incomplete.

One of the intentions of this work is to exploit more powerful tools to solve such ambiguous cases by using a richer symbolic description. In Section 4 we will give an example.

3. Knowledge Representation

Previous work on perceptual anchoring did not pay much attention to the nature of the symbolic system that is tied to the anchoring process, with the notable exception of [2]. The symbolic system described in the framework of Section 2 assumes a set of symbols denoting objects of interest and a set of not further specified (unary) predicates stating properties of these objects. A drawback of that approach is that the framework does not pose any restrictions on, and does not provide any means to organise and manage that symbolic knowledge, or to use it for reasoning and inference of new knowledge.

The main contribution of this work is the use of a symbolic knowledge representation and reasoning (KRR) system as the symbolic component in the anchoring framework. By providing the anchoring process with a knowledge base described in a Description Logic [1] we expect to solve simple cases of ambiguity by using conceptual and semantic knowledge about the anchored objects and that reasoning and inference with a set of carefully designed rules can provide a powerful tool to resolve conflict cases. In addition, by using a formal way to represent the knowledge it can be easier exchanged and used by other parts of the system, and if designed properly can facilitate communication with human users.

3.1. The Knowledge Base

The conceptual knowledge is structured in a hierarchy, a so called *ontology*, allowing definition of concepts at different levels of abstraction and supporting subsumption inference. An ontology specifies an abstract and simplified view of the world (the domain) that we want to model, and can be, at least partially and up to a certain level of detail, defined independently from a specific application. Practically, an ontology constitutes an agreement to use a shared vocabulary and constraints on the interpretation of terms, that is consistent with the modeled domain. A major advantage of the use of an ontology is that knowledge in a knowledge base can be exchanged between agents (including humans) without depending on an interpretation context and that it can be easily queried.

The knowledge base consists of two parts: a *terminological component* (called T-Box) that contains the description of the relevant concepts and their relations; and an *assertional component* (called A-Box) storing concept instances and assertions on those instances. Keeping both parts separate is convenient, to maintain the distinction between conceptual knowledge, which is mostly static and largely independent of an actual anchoring scenario, and the assertions actually concerning a scenario, which might be of very dynamic nature.

3.2. The Domain Ontology

For our domain, the anchoring problem, we require an ontology that covers all the physical entities and their perceptual properties that can occur in an anchoring scenario, and thus are recognised by the perceptual system. In addition we want to use knowledge that can be inferred from basic knowledge about anchors or that is collected from external sources with cognitive capabilities, such as other anchoring processes, or in particular humans interacting with the system. Modelling an ontology is in general a difficult task, and is not our direct concern; therefore we chose to base our ontology on a subset of the ontology framework DOLCE (A Descriptive Ontology for Linguistic and Cognitive Engineering) [13], an upper-level ontology developed for the Semantic Web with an emphasis on cognitive aspects.

The main concepts in DOLCE are divided into the categories Endurants, Perdurants, Qualities, and Abstracts. Endurants are entities that are present at whole (including all their proper parts) at any time they are present (for example natural objects, like cups, or other agents), whereas Perdurants are only partially present, their parts evolving and unfolding over time (for example an event). Qualities describe basic entities that can be perceived and measured by agents. DOLCE makes a distinction between the quality of an entity which is a concept inherent to that entity, and the actual value of that quality, its *quale* (often called property). This stems from the idea of a *conceptual space* [10].

Figure 1 shows an excerpt of our domain ontology for the anchoring problem used in this work. Objects known to the anchoring module are sub-concepts of Physical Object and can have a number of qualities (color, smell, size), defined by the leaves of the Quale hierarchy. In this work, we do not employ the concept Quality to represent properties and their values, but we use the property values provided through the grounding relation of the anchoring module.

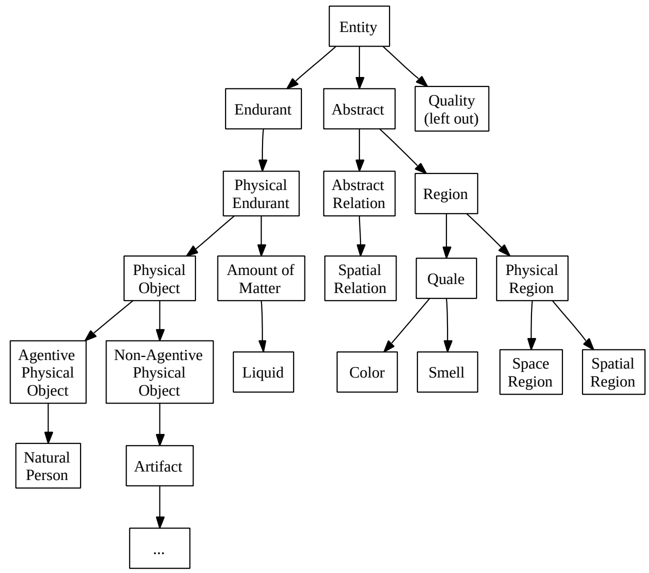

Figure 1. An excerpt of the used ontology for perceptual anchoring, based on DOLCE [13].

3.3. Rules

Another aspect of many KRR systems is the possibility to define rules that trigger actions when facts are added, removed, or changed in the A-Box, or when the subsumption inference classifies an instance as being of a more specific type. We try to make use of some rules that provide extended inference capabilities beyond the simple concept-based ones provided by the T-Box reasoner of the chosen KRR system; see the next section for an example.

4. Implementation

The anchoring framework is implemented in LISP and is connected to a suitable perceptual system. The anchoring module is integrated into the robot control architecture and makes its functionalities and established anchors available to other parts of the system. For example, a high-level task planner that operates on the anchors symbolic description, or the low-level behavioural control system, that uses the perceptual signature of anchors to navigate to objects. The knowledge base is implemented using the Loom knowledge representation system [3], running in a separate LISP process and hooked to the anchoring module through a middleware software (see Figure 2).

The experimental test-bed for our system is a mobile robot platform that is part of an ambient intelligent environment, called the PEIS Ecology [17]. The robot "shares" a small furnished apartment with humans and other ambient

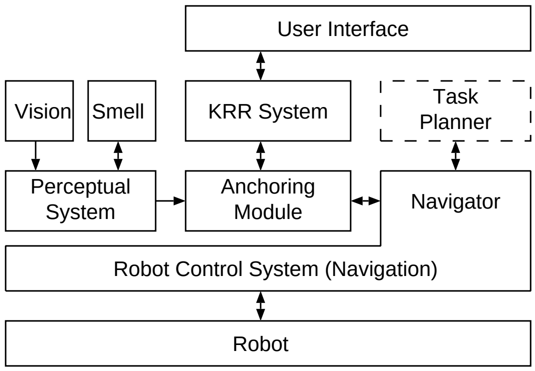

Figure 2. Schematic overview of the system parts and their connections.

intelligent devices, and is able to exchange information with these devices through the middle-ware software. The perceptual system of the robot consists of a vision system that can detect a set of object shapes and properties like color, relative size, and an object's location with respect to the robot, and an olfactory system (an electronic nose with a trained classifier) that can identify a small set of odors in close proximity to the robot, when triggered by the system.

All the following experiments were done in simulation, using the Player/Stage software[1] and two helper programs that provide the visual percepts and smell information from the simulation. We want to point out that it technically does not make a difference for the system if it is run in simulation, apart from perfect conditions and the absence of perceptual failures, but only that the sensor data for the Player server is provided by the Stage simulation module.

4.1. Simple Inferences

For each object, when it is anchored, it's symbolic description is created in the KB: an instance of the respective concept (object class) is created with the given properties, like it's color. A first advantage of keeping the symbolic description of anchors in a knowledge base is the fact that the descriptions used to find anchors can be much richer, and can include inferable knowledge.

So far, we added a simple hierarchy of drinking vessels to the ontology, defined as follows (in Loom):

```
(defconcept Vessel
    :is-primitive Artifact)
(defconcept Drinking-Vessel
    :is-primitive Vessel)
(defconcept Mug
    :is (:and Drinking-Vessel
              (:filled-by Has-Size 'SMALL)))
(defconcept Has-Handle)
```

[1] Available from http://playerstage.sourceforge.net

```
(defconcept Cup
    :is (:and Mug Has-Handle))

(defconcept Color
   :is-primitive Quale)
(defrelation Has-Color
   :domain Physical-Object
   :range  Color)
(defconcept Size
   :is-primitive Quale)
(defrelation Has-Size
   :domain Physical-Object
   :range  Size)
```

The vision system can identify mugs and cups, with cups having a handle; the handle is treated as a property of the cup and is not a separate physical entity. This avoids the problem of anchoring compound objects, that we do not want to address here.

In our scenarios, mugs and cups, and though more general, vessels can contain liquids, that we consider to be of class *Amount-Of-Matter*; the concept of a liquid does not exist without it's container, therefore we introduce a dependency on the respective relation *Contains-Liquid*:

```
(defconcept Liquid
    :is (:and Amount-Of-Matter
              (:exactly 1 Contains-Liquid)))
(defrelation Contains-Liquid
    :domain Vessel
    :range  Liquid)
```

To make use of the odor classification which the robot provides, we add a smell property that is inherent to the class *Amount-Of-Matter*, and a second relation asserting the smell of an object (*Smells-Of*):

```
(defconcept Smell
    :is (:and Physical-Quality
              (:at-least 1 Has-Smell)))
(defrelation Has-Smell
    :domain Amount-Of-Matter
    :range  Smell)
(defrelation Smells-Of
    :is (:satisfies (?x ?y)
          (:or (Has-Smell ?x ?y)
               (:for-some (?z)
                 (:and (Contains-Liquid ?x ?z)
                       (Has-Smell z? ?y))))))
```

Now, given for example the facts *Liquid(coffee-liquid), Smell(coffee-smell), Has-Smell(coffee-liquid, coffee-smell)*, and the assertion *Smells-Of(mug, coffee-smell)* we want the system to infer that the mug contains a liquid smelling of coffee, say coffee, which we consider the only reasonable explanation in our scenario. Such an inference is not possible given the description above, and in fact this is an abduction inference, which is not possible per-se in the KRR system. We decided to solve this with a handcrafted rule *Assert-Smells-Of* (in Loom):

```
(defmethod Assert-Contains-Liquid (?x ?y)
    :situation (:and (Vessel ?x) (Smell ?y))
    :with      (:and (Liquid ?z) (Has-Smell ?z ?y))
    :response  ((tellm (Contains-Liquid ?x ?z))))
(defproduction Assert-Smells-Of
    :when    (:and (Vessel ?x) (Smell ?y)
                   (:detects (Smells-Of ?x ?y)))
    :perform (Assert-Contains-Liquid ?x ?y))
```

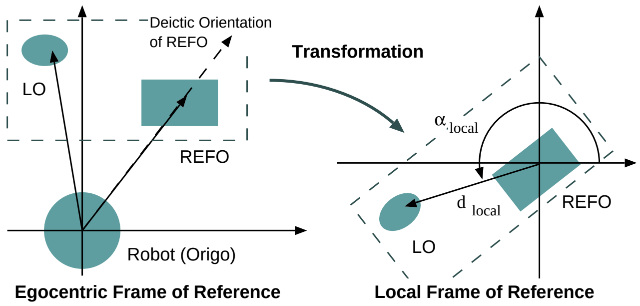

Figure 3. Frame of reference, and computation of distance and direction angle. Objects are represented by their idealised point location.

Whenever a *Smells-Of* assertion is added to the A-Box of the KB, the method *Assert-Contains-Liquid-Has-Smell* asserts that the respective vessel contains the liquid with that smell (assuming that there is only one such).

Example run: We drove the robot manually around in the simulated environment, approaching an object that was identified as a mug by the vision system and anchored; while moving the robot around the object, a handle was detected, which automatically classified the mug as a cup, after the fact was added to the KB. Going close to the cup, a smell action was manually triggered, which asserted that the cup smelled of coffee. This in turn fired the above rule, adding the fact that the cup contained coffee to the KB. A query on the KB for a cup containing coffee returned the previously anchored object.

4.2. Spatial Relations

Spatial relations are used in the symbolic description of objects and allow to distinguish objects by their location w.r.t. other objects, and play an important role when it comes to human-robot interaction. Two classes of binary spatial relations between a reference object $REFO$ and the located object LO are considered: the distance (topological) relations "at" and "near", and the directional (projective) relations "front of", "behind", "right", and "left". The interpretation of a projective relation depends on a frame of reference; for reasons of simplicity we assume a deictic frame of reference with an egocentric origin coinciding with the robot platform. Similar to the approach in [12], we model spatial relations as concepts in the ontology (see Section 3.2): we consider a spatial relation a sub-concept of Abstract Relation (itself a subclass of Abstract), having as properties a reference object, a located object (both Physical Objects; the origin is omitted in this implementation), and a spatial region, an instance of the Abstract concept Spatial Region, which is one of the six defined (in Loom):

```
(defconcept Abstract-Relation
    :is-primitive Abstract)
(defconcept Spatial-Relation
    :is (:and Abstract-Relation
              (:exactly 1 Has-Reference-Object)
              (:exactly 1 Has-Located-Object)
              (:exactly 1 Has-Spatial-Region)))
(defconcept Spatial-Region)
(tellm (create AT Spatial-Region)
       (create NEAR Spatial-Region)
       (create LEFT Spatial-Region)
       (create RIGHT Spatial-Region)
       (create BEHIND Spatial-Region)
       (create IN-FRONT Spatial-Region))
(defrelation Has-Reference-Object
    :domain Spatial-Relation
    :range  Physical-Object)
(defrelation Has-Located-Object
    :domain Spatial-Relation
    :range  Physical-Object)
(defrelation Has-Spatial-Region
    :domain Spatial-Relation
    :range  Spatial-Region)
```

For the computation and evaluation of these spatial relations we use the model presented in [9] and apply it to 2D space. The evaluation of a spatial relation results in a degree of applicability, in a range between "not" and "fully" applicable, respectively. A local coordinate system at the $REFO$, aligned to its deictic orientation, as shown in Figure 3, is defined, and the local coordinates of the LO w.r.t. the $REFO$ are computed. From this the Euclidean distance $d_{\text{local}}(LO)$ and the direction angle α_{local} are computed.

We use simple trapezoidal membership functions μ_{topo} and μ_{proj} for the evaluation (others are possible, e.g. spline functions):

$$\begin{aligned} a_{topo} &: (LO, REFO) \mapsto \mu_{topo}(d_{\text{local}}(LO)) \\ a_{proj} &: (LO, REFO) \mapsto \mu_{proj}(\alpha_{\text{local}}(LO)) \end{aligned}$$

with $topo \in \{at, near\}$ and $proj \in \{front, behind, left, right\}$ that partition the space in qualitative acceptance areas. Figure 4 shows a possible definition of the functions μ_{topo} (top) and μ_{proj} (bottom).

The computation of spatial relations can be triggered on behalf and can be restricted to a set of anchors, usually those that are relevant in the current context. The algorithm computes all possible spatial relations between all given anchors and selects the resulting set of applicable relations, those relations with a degree of applicability above a predefined threshold. For each of the selected relations an instance of Spatial Relation is created in the knowledge base with the corresponding reference and located objects, and Spatial Region.

To allow spatial references to objects from an egocentric perspective of the robot, we define a special anchor named *Me* that is always located at the origo of the reference frame, with position $(0, 0)$. For example, the robot now can process queries like "the red ball left of you" assuming that the system relates the reference "you" to the anchor *Me*.

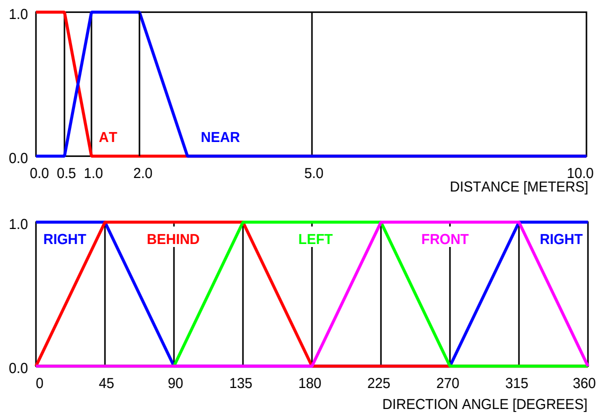

Figure 4. Evaluation functions for topological and projective relations.

An example of the use of spatial relations is given in Section 4.4.

4.3. Human-Robot Interaction (HRI)

The user-interface to the robot consists of a plain text based application, where the user can type in sentences in very simple English. The sentence is analysed by a recursive descent parser and translated into a symbolic description. The grammar allows commands of the form "find ..." following a description of the object. The description consists of a main part and can be followed by sub-clauses describing objects that are spatially related to that object. The main part and each of the sub-clauses can be either a definite or indefinite description, indicated by the article "a" or "the", and includes the object's class, for example "cup", and optionally it's color and smell. The smell of an object is inferred from the clause "with ..." following the object's class, indicating that it contains a liquid; for example, "the cup with coffee" is assumed to be a cup containing coffee, and as such, smelling of coffee.

The derived symbolic description is used to construct a query for the KB. The main functionality is realised by a FIND-ANCHOR routine, which collects candidates from the KB that match the description. It triggers the computation of spatial relations for the established anchors, and starts by collecting candidates for the main object using it's shape (class) and optional color; for each of the sub-clauses with a spatial relation, it collects candidates that match the description (again shape and optional color) and that are the reference object of a spatial relation with the specified region and the main object as located object. The final set of candidates is the intersection of all collected candidates.

If there is more than one candidate, the anchoring module checks for further properties in the given description, apart from shape and color, and selects the property smell if specified. In case there are no facts about the smell of the candidate objects in the KB, the system triggers a smell action for each object. This results in the assertion of *Smells-Of* facts. The candidate set is filtered with respect to the specified smell (say, each candidate not part of the *Smells-Of* relation with that smell is discarded). If the set of candidate objects is still not unique, the description given by the user is ambiguous and not sufficient to find the object.

In a last attempt, the system tries to ask the user about which of the candidate objects to select. It does so by describing their spatial layout with respect to the robot's view point. It selects the projective relations that have the candidate objects as located objects and the special anchor *Me* as reference object, constructs a sentence that enumerates the objects with their spatial location, and prompts the user for an answer. The user has to reply with a "find" command specifying the target object and its spatial location, resulting in an additional symbolic description. The candidates are filtered with the query generated from that description. If this does not lead to a single candidate, the system fails and informs the user that it is unable to find the requested object.

4.4. Example Run

Having a scenario in mind, where the user sends his service robot into the kitchen to fetch the user's cup with coffee, we presented the simulated robot with a static setup of two white cups and a red bottle, placed in front of it. The vision module identified the objects, and they were correctly anchored, with the respective information created in the KB. The user typed in the following sentence:

"Find the white cup with coffee at the red bottle"

In a first step a query for white cups returned the two candidates, and the use of the spatial relation did not resolve that ambiguity as both cups were at the red bottle. In the next step information about the smell of both cups was collected as the KB did not contain any such information. The smell actions asserted that both cups smelled of coffee, and though contained coffee, which did not resolve the ambiguity. In the last step the system asked the user which of the cups to select by constructing the following question:

"There is more than one white cup with coffee at the red bottle: a cup left of me and a cup right of me. Which one do you mean?"

The user answered *"The cup left of you"*, which after filtering the candidate objects with the generated query, lead to a single object, the cup to the left of the robot.

5. Discussion

Various aspects of the anchoring problem, as outlined in Section 2 have been studied within the literature. We consider our work to be the first to use a knowledge representation and reasoning system for the anchoring process.

5.1. Related Work

Work involving formal knowledge representation for perceptual anchoring was done by Bonarini et al. [2] for the domain of robotic soccer. They give a general description of a knowledge representation model based on the notion of concept and its properties. Concepts are defined by their inherent properties and are organized in a concept hierarchy according to their super- and sub-concept relations. No details about an actually implemented knowledge base or possible reasoning capabilities, apart from concept classification provided through the concept hierarchy, are reported.

Mastrogiovanni et al. [14] describe a symbolic data fusion system for an ambient intelligent environment, consisting of several cognitive agents with different capabilities, according to the extended JDL (Joint Directors of Laboratories) model (from sensor and data fusion to situation and impact assessment). The lowest level consists of the network of (virtual) sensors, that provide a second level with percepts that are then fused symbolically. Although not explicitly mentioned, the second level performs anchoring using a knowledge base in Description Logic. The system is capable of simple straight forward inferences and is basically used for data interpretation, which is in the next higher level used for action detection and situation assessment.

From a different perspective, the anchoring problem can be seen as a richer but not that encompassing and challenging version of the problem that the Cognitive Vision community is tackling. At its core, each cognitive vision system has to solve the anchoring problem somehow. The work presented by Hois et al. [12] considers the problem of integrating spatial relations into an domain ontology for a robot platform equipped with a 3D LASER scanner that observes static scenes in an office environment. The ontology helps to classify the detected objects, and in a second stage, the user can query the system for simple object identification and localisation tasks, involving spatial relations.

So far, the use of spatial relations for anchoring has not been studied in detail. Earlier work [4], investigating the use of planning techniques to recover from perceptual failures and ambiguous cases in perceptual anchoring, incorporated a simple means to refer to an object by specifying its relations to other anchored objects, but only the relations "at" and "near" where supported, and computed on-the-fly using a simple and crisp computational model. Our previous work [15] showed a first implementation of spatial relations that could be used in the symbolic description and in a simple scripted human-robot-dialogue to resolve an ambiguous description, but without the support of a knowledge base.

5.2. Future Work

The implemented system is an ad-hoc design and the coupling between the anchoring module and the knowledge base is rather loose. In a first step we want to tighten the integration of the KRR system, making it the primary symbolic interface to the anchoring module, and possibly providing a query language (or data structure) that is compatible with the other symbolic parts of the system, like the task-planner or the user-interface.

The linguistic HRI part is still very rudimental and based on a text interface, and reminds of the capabilities of Winograd's SHRDLU system [18]. We intend to use a simple speech dialogue system in future work, similar to the system of Hois et al. [12]. In such systems, the human-robot interaction is limited to a (conventional) "master-slave" mode of communication, but our interest is to enable the robot to make use of humans in order to compensate for perceptual or cognitive deficiencies. A good example in this line of thought is the "Peer-to-Peer Human-Robot Interaction" project [8], that aims to develop a range of HRI techniques so that robots and humans can work together in teams and engage in task-oriented dialogue.

Many aspects of the example in Section 4.3 rely on the rigid implementation of the FIND-ANCHOR functionality, which in itself is complicated and not yet flexible enough. We intend to investigate the use of a planner, or planning techniques, to decide on the different steps and choices in the recovery/disambiguation process and to coordinate them in a more coherent way. We are particularly interested in using such techniques to navigate through the knowledge base, and to guide the selection of relevant information, also with respect to the HRI part. An interesting starting point is provided in [19]. As encountered in Section 4.3 abductive reasoning is a necessary requirement for the system to be able to handle cases where inference is not possible. This is very much related to formal frameworks for knowledge acquisition and belief revision in agents (see for example [16]).

6. Conclusion

In this work we have extended the anchoring framework with a knowledge representation and reasoning (KRR) system for the symbolic part of the anchors. We consider our work to be the first to use a KRR system for the anchoring process. A first implementation provided an ontology and a knowledge base (KB) for storing a set of objects and properties, and spatial relations between those objects. The

KB makes is easier to manage the information and queries on the anchored objects can be more advanced and are processed by the KB. The given example, in which the system resolves an ambiguity by gathering perceptual information and finally involving the user in the anchoring task, reveals starting points for future work. In a first step we want to refine the design and implementation of the KB integration and investigate how planning techniques can be used to coordinate the disambiguation process and the human-robot interaction part. Another point of interest is the use of abduction inference for belief revision.

References

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

[2] A. Bonarini, M. Matteucci, and M. Restelli. Concepts for anchoring in robots. In *Proceedings of AI*IA 2001 – 7th Congress of the Italian Association for Artificial Intelligence*, pages 327–332, Bari, Italy, 2001.

[3] D. Brill. Loom reference manual, for Loom version 2.0. Technical report, ISI, University of Southern California, USA, 1993.

[4] M. Broxvall, S. Coradeschi, L. Karlsson, and A. Saffiotti. Recovery planning for ambiguous cases in perceptual anchoring. In *Proceedings of AAAI 2005 – 20th National Conference on Artificial Intelligence*, Pittsburgh, USA, 2005.

[5] S. Coradeschi and A. Saffiotti. Anchoring symbols to sensor data: Preliminary report. In *Proceedings of AAAI-00 – 17th National Conference on Artificial Intelligence*, Austin, USA, 2000.

[6] S. Coradeschi and A. Saffiotti. An introduction to the anchoring problem. *Robotics and Autonomous Systems*, 43:85–96, 2003.

[7] S. Coradeschi and A. Saffiotti. Symbiotic robotic systems: Humans, robots, and smart environments. *IEEE Intelligent Systems*, 21(3):82–84, 2006.

[8] T. Fong, I. Nourbakhsh, R. Ambrose, R. Simmons, A. Schultz, and J. Scholtz. The peer-to-peer human-robot interaction project. In *Proc. of the AIAA Space 2005 Conf.*, Long Beach, CA, USA, 2005.

[9] K.-P. Gapp. An empirically validated model for computing spatial relations. In *Proceedings of KI-95 – 19th Annual German Conference on Artificial Intelligence*, Bielefeld, Germany, 1995.

[10] P. Gärdenfors. *Conceptual Spaces: The Geometry of Thought*. MIT Press, 2000.

[11] D. Hernández. *Qualitative Representation of Spatial Knowledge*, volume 804 of *LNCS*. Springer-Verlag, 1994.

[12] J. Hois, M. Wünstel, J. Bateman, and T. Röfer. Dialog-based 3D-image recognition using a domain ontology. In *Proc. of the Int. Conf. Spatial Cognition 2006*, Bremen, Germany, 2006.

[13] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, and L. Schneider. The wonderweb library of foundational ontologies. Technical Report WonderWeb Deliverable D17, ISTC-CNR, Padova, Italy, 2003.

[14] F. Mastrogiovanni, A. Sgorbissa, and R. Zaccaria. A distributed architecture for symbolic data fusion. In *Proceedings of IJCAI-07 – International Joint Conference on Artificial Intelligence*, Hyderabad, India, 2007.

[15] J. Melchert, S. Coradeschi, and A. Loutfi. Spatial relations for perceptual anchoring. In *Proceedings of AISB'07 – AISB Annual Convention*, Newcastle upon Tyne, UK, 2007.

[16] M. Pagnucco. *The Role of Abductive Reasoning within the Process of Belief Revision*. PhD thesis, Basser Department of Computer Science, University of Sydney, Australia, 1996.

[17] A. Safiotti and M. Broxvall. PEIS Ecologies: Ambient intelligence meets autonomous robotics. In *Proceedings of sOc-EUSAI '05 – Joint Conf on Smart Objects and Ambient Intelligence*, Grenoble, France, 2005.

[18] T. Winograd. Procedures as a representation for data in a computer program for understanding natural language. Technical Report AITR-235, MIT Artificial Intelligence Laboratory, Boston, MA, USA, 1971.

[19] J. L. Wyatt. Planning clarification questions to resolve ambiguous references to objects. In *Proceedings of the 4th Workshop on Knowledge and Reasoning in Pra ctical Dialogue Systems, IJCAI-05 – International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland, 2005.