

International Journal on Artificial Intelligence Tools  
© World Scientific Publishing Company

## Using Knowledge Representation for Perceptual Anchoring in a Robotic System

Amy Loutfi, Silvia Coradeschi, Marios Daoutis, Jonas Melchert

*Center for Applied Autonomous Sensor Systems  
Örebro University  
70182 Örebro, Sweden  
Email: amy.loutfi@tech.oru.se*

In this work we introduce symbolic knowledge representation and reasoning capabilities to enrich perceptual anchoring. The idea that encompasses perceptual anchoring is, the creation and maintenance of a connection between the symbolic and perceptual description that refer to the same object in the environment. However, without higher level reasoning, perceptual anchoring is still limited. Hence we direct our focus to combining a knowledge representation and reasoning (KRR) system with the anchoring module to exploit a knowledge inference mechanisms. We implemented a prototype of this novel approach to explore through elementary experimentation the advantages of integrating a symbolic knowledge system to the anchoring framework in the context of an intelligent home. Our results show that using the KRR we are better able to cope with ambiguities in the anchoring module through exploitation of human robot interaction.

### 1. Introduction

An emerging trend in the field of robotics is the notion of *symbiotic robotic systems* which consists of a robot, human and (smart) environment cooperating together in performing different tasks<sup>5</sup>. By assisting the robot with information provided by the human or smart objects, some of the current challenges in robotics can be circumvented. For instance, localization of the robot can be done with a system of surveillance cameras and object recognition tasks can be assisted by passive technologies like RFID. Human assistance and cooperation can also be used to provide instructions to the robot and to assist the robot in case of failure or ambiguous situations when several choices are possible. The motivation behind the symbiotic system is the integration of robotics into everyday life. Therefore, it is essential to allow a range of different users to be able to communicate to the system, this range should include both expert users and even bystanders.

A natural form of communication between humans and the robots is natural language dialogue. Amongst the many challenges that this task presents, in this paper, we concentrate on the correspondence that must necessarily exist between the linguistic symbols used by a human and the sensor data perceived by the robot. We call *anchoring* the process of creating and maintaining over time the connection between the symbols and the corresponding perceptual representation that refer to the same physical objects. Already in the field of robotics, anchoring has been explored in systems that use planning and a variety of sensing modalities (e.g. vision and olfaction). In this paper we examine the possibility to integrate the anchoring framework in a symbiotic robotic system which includes a knowledge

2 Amy Louf, Silvia Coradeschi, Marios Daoutis, Jonas Melchert

representation and reasoning component. We argue that this integration is an important ingredient to obtaining a natural and effective interaction, particularly in cases where a robot may assist the human in simple tasks. In this paper, the use of the KRR is further advocated to also allow the human to assist the robot in simple anchoring tasks, such as the disambiguation of objects, thereby exploring a deeper form of mutual human robot interaction (HRI).

The validation of the system is done in the context of an intelligent home environment, which can be used for ambient assisted living for the elderly or disabled. In this intelligent home, an existing framework called the PEIS-Ecology is used to coordinate the exchange of information between the robot, other pervasive technologies in the environment and the human user. To illustrate the utility of KRR in anchoring three case studies are presented. The first focusses on the inclusion of spatial relations in the anchoring framework. A set of binary spatial relations, “at”, “near”, “left”, “right”, “in front”, and “behind” are used for 2D space. As spatial prepositions are inherently rather vague, we are using fuzzy sets to define graded spatial relations. The proposed method computes a spatial relations-network for anchored symbols and stores that in the KRR. A second case uses multi-modal information about objects which includes both spatial information and in this case, olfactory information. The KRR is then used to assist the robot in determining which perceptual actions can be taken to collect further information about object properties. The third case investigates the possibility to reason about object properties in order to determine optimal candidate selection. Each of the case studies have been selected based on novelty of our approach and the relevance for an intelligent home application.

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 and shows its functioning in Section 5. Section 6 discusses our approach with respect to related work with a final conclusion and mention of future work in Section 7.

## 2. Perceptual Anchoring

As described in the introduction, the task of anchoring is to create and maintain in time the correspondence between symbols and percepts that refer to the same physical object. This correspondence is reified in a data structure  $\alpha(t)$ , called an *anchor*. It is indexed by time as the perceptual system continuously generates new percepts; and the created links are dynamic, since the same symbol may be connected to new percepts every time a new observation of the corresponding object is acquired. So at each time instance  $t$ ,  $\alpha(t)$  contains a symbol identifying that object, a percept generated by the latest observation of the object, and a perceptual signature meant to provide the (best) estimate of the values of the observable properties of the object.

The main parts of anchoring are<sup>4</sup> :

- A *symbol system* including a set  $\mathcal{X} = \{x_1, x_2, \dots\}$  of individual symbols (variables and constants), a set  $\mathcal{P} = \{p_1, p_2, \dots\}$  of predicate symbols, and an infer-

ence mechanism whose details are not relevant here.

- A *perceptual system* including a set  $\Pi = \{\pi_1, \pi_2, \dots\}$  of possible percepts, a set  $\Phi = \{\phi_1, \phi_2, \dots\}$  of attributes, and perceptual routines whose details are not relevant here. A percept is a structured collection of measurements assumed to originate from the same physical object; an attribute  $\phi_i$  is a measurable property of percepts with values in the domain  $D(\phi_i)$ . Let  $D(\Phi) = \bigcup_{\phi \in \Phi} D(\phi)$ .
- A *predicate grounding relation*  $g \subseteq \mathcal{P} \times \Phi \times D(\Phi)$ , which embodies the correspondence between (unary) predicates and values of measurable attributes. The relation  $g$  maps a certain predicate to compatible attribute values.

The following definitions allow to characterize objects in terms of their (symbolic and perceptual) properties:

- A *symbolic description*  $\sigma$  is a set of unary predicates from  $\mathcal{P}$ .
- A *perceptual signature*  $\gamma : \Phi \mapsto D(\Phi)$  is a partial mapping from attributes to attribute values.

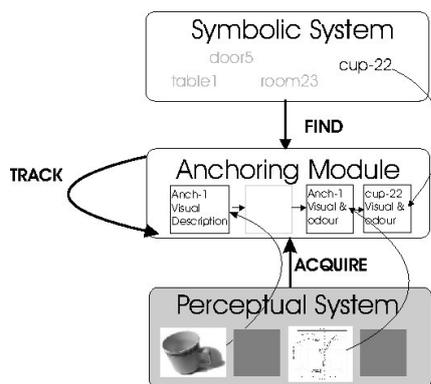


Fig. 1. Graphical illustration of the anchoring framework: the anchoring module connects the perceptual and the symbolic systems in a physically embedded intelligent system. The perceptual system consists of visual and olfactory percepts.

## 2.1. Creation and Maintenance of Anchors

The extension of the framework previously presented allows the creation of anchors in both a top-down and a bottom-up fashion: bottom-up acquisition is triggered by recognition events from the sensory system when percepts can not be associated with existing anchors; top-down acquisition occurs when a symbol needs to be anchored to a perceptual description (such a request may come from a top-level planner)<sup>11</sup>. These functionalities are realized through an *acquire* and *find* functionality. See Fig 1 for a graphical illustration.

At each update cycle of the perceptual system, when new perceptual information is received, it is important to determine if the new information should be associated to an

4 Amy Loufi, Silvia Coradeschi, Marios Daoutis, Jonas Melchert

existing anchor (data association problem). A *track* functionality addresses the problem of tracking objects over time. Furthermore for specific cases of top-down anchoring two important features should be highlighted:

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

**Definite versus Indefinite Descriptions:** The given symbolic descriptions can be either *definite* or *indefinite*: a description is definite if it denotes a unique object, 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, for example “a coffee-cup”.

Given a request in the form of a symbolic description and information about whether this is a definite or indefinite description, the anchoring module has to find a matching candidate and has to detect possible ambiguities. If an unambiguous match is found, the anchoring module selects that candidate for anchoring. In the case if there are ambiguities and/or no fully matching candidates are found, the anchoring process could make use of a high-level task planner to create and execute a recovery plans with the aim of searching for unperceived objects or collecting more perceptual properties<sup>3</sup>. In order to do this, higher level knowledge about which properties are relevant for disambiguation is needed. Furthermore, in cases where ambiguities cannot be resolved by active perception, higher level knowledge about objects and the behaviour of objects may also be used. The intention of this work is to exploit more powerful tools to solve such ambiguous cases by using a richer symbolic description.

### 3. Knowledge Representation

The conceptual knowledge that we are using 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 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 modelled 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 in order to maintain the distinction between conceptual knowledge and the assertions actually concerning a scenario. The con-

ceptual knowledge is mostly static and largely independent of an actual anchoring scenario, whereas the assertions might be of very dynamic nature.

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) <sup>13</sup>, an upper-level ontology developed for the Semantic Web. From the possible options we selected DOLCE because it suits our needs and is comparably simple.

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* <sup>8</sup>.

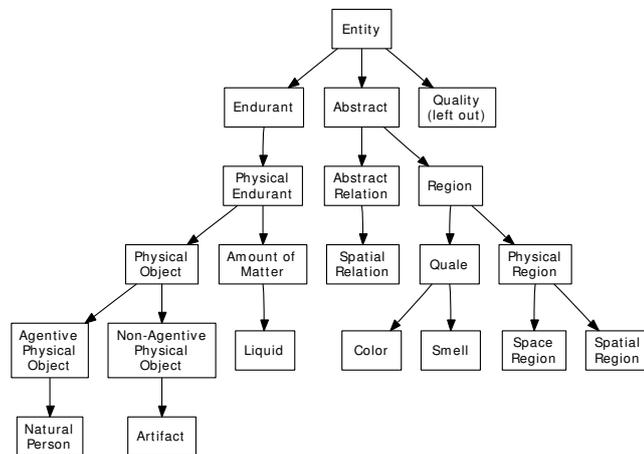


Fig. 2. An excerpt of the used ontology for perceptual anchoring, based on DOLCE.

Fig 2 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 (colour, 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 use the symbolic property values delivered by the perceptual system, which

6 *Amy Loufi, Silvia Coradeschi, Marios Daoutis, Jonas Melchert*

are explicitly mapped as instance properties to the KB and uses a hand-crafted grounding relation, delivering percepts of the form:

```
((OBJECT (ID 1) (SHAPE MUG) (COLOR WHITE)
  (POSITION (1400 400))))
```

Another aspect of most 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 <sup>2</sup>, running in a separate LISP process and hooked to the anchoring module through a middle-ware software further explained in Section 5.

##### 4.1. Simple Inferences

For each object, when it is anchored, its symbolic description is created in the KB: an instance of the respective concept (object class) is created with the given properties, like its colour. 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)
(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 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 its 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 odour classification that 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
        (:exactly 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, 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 that is not possible per se in the KRR system. For this reason we had to explicitly map each substance, to each substance-smell.

```
(tellm
  (Has-Smell coffee-liquid coffee-smell)
  (Has-Smell ethanol-liquid ethanol-smell)
  (Has-Smell hexanal-liquid hexanal-smell)
  (Has-Smell octanol-liquid octanol-smell)
  (Has-Smell 3-hexanal-liquid 3-hexanal-smell)
  (Has-Smell linalool-liquid linalool-smell)
  ...
  ...)
```

8 Amy Loufi, Silvia Coradeschi, Marios Daoutis, Jonas Melchert

)

Then using a hand-crafted production rule *Assert-Contains-Liquid-Has-Smell*, we achieved this simple inference mechanism (in LOOM):

```
(defproduction Assert-Contains-Liquid-Has-Smell (?x ?y)
 :situation (:and (Vessel ?x) (Smell ?y))
 :with      (:and (Liquid ?z) (Has-Smell ?z ?y))
 :response  ((tellm (Contains-Liquid ?x ?z))))
```

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

#### 4.2. Spatial Relations

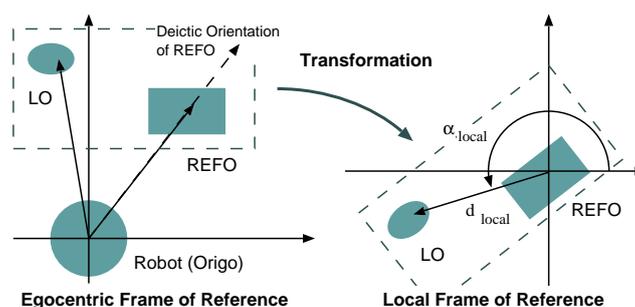


Fig. 3. Frame of reference, and computation of distance and direction angle.

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 <sup>10</sup>, we model spatial relations as concepts in the ontology (see Section 3): 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 Gapp<sup>7</sup> 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 Fig 3, is defined, and the local coordinates of the *LO* w.r.t. the *REFO* are computed. From this the Euclidean distance  $d_{\text{local}}$  and the direction angle  $\theta_{\text{local}}$  are computed.

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

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

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

The computation of spatial relations can be triggered on command 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 to the left of you” assuming that the system relates the reference “you” to the anchor *Me*.

An example of the use of spatial relations is given in Section 5.2.

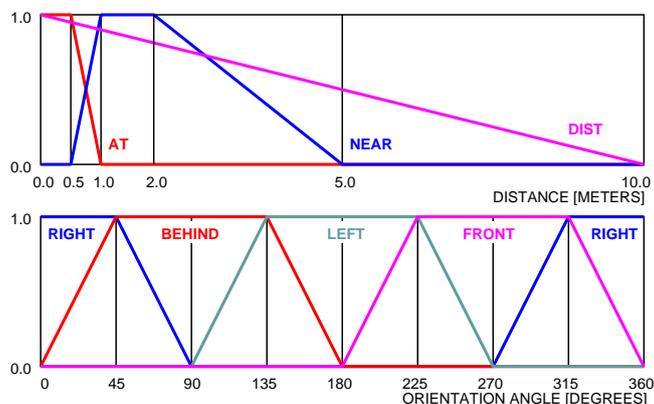


Fig. 4. Evaluation functions for topological and projective relations.

### 4.3. Managing Object Properties

A specially interesting case is one in which the importance of an object's property may vary depending on context. Consider the example of an intelligent home where a request is made to a robot to retrieve a specific fruit. In general normally fresh fruits are suitable candidates. Knowing that certain fruits, such as bananas, may have a varying color property depending on its freshness, this information can be used to determine the suitability as candidate anchor, without requiring the user to make a detailed request for a 'yellow fresh banana'.

To handle this type of information in the KB, we extend our ontology's hierarchy to define concepts representing objects (e.g. fruits), where these objects are a sub-class of the physical objects domain, having properties such as *colour*, *size*, *shape*, *age* and *name* (in LOOM) such as:

```
(defconcept Fruit :is-primitive
  (and Physical-Object
    (:exactly 1 Has-Name)
    (:exactly 1 Has-Colour)
    (:exactly 1 Has-Size)
    (:exactly 1 Has-Shape)
    (:exactly 1 Has-Fruit-Age)))
```

In this way, each anchor that comes from sensory systems creates an instance of the concept fruits to the ABox. The spatial reasoning component remains as it is, since it is designed to function properly in every situation, whether we try to find cups, or fruits. In the implementation, the temporal aspect is considered as a set of states rather than linear time temporal logic (LTL). Although LOOM is capable of linear time temporal reasoning, we find that the "state" approach efficiently interconnects with the spatial component which is designed in description logic.

Here, we introduce the concept "age" for fruits, which may be assigned with specific values from the one-way state set. We do not allow a fruit to change states backwards :

```
(defset Fruit-Age :is ( :the-ordered-set 'FRESH 'MATURE 'ROTTEN))

(defrelation Has-Fruit-Age
:domain Fruit
:range Fruit-Age
:characteristics :single-valued)
```

Then we define some additional relations, that would help the KB to classify each fruit instance that is anchored, with the appropriate age. With respect to TBox specification:

```
Fresh-Banana :is ( :or (the Has-Fruit-Age (the-set 'FRESH))
                       (the Has-Colour (the-set 'YELLOW)))

Mature-Banana :is ( :or (the Has-Fruit-Age (the-set 'MATURE))
                       (the Has-Colour (the-set 'BROWN)))

Rotten-Banana :is ( :or (the Has-Fruit-Age (the-set 'ROTTEN))
                       (the Has-Colour (the-set 'BLACK)))
```

So the KB can tell about the age of the bananas based on their colour, or their previous (if any) classification. To be able to prevent returning in-consumable *Fruits*, we then define a production rule which consistently checks for *Rotten-Fruits* instances and after announcing their age, it removes them from the ABox since we do not desire our service robot to return such instances :

```
(defproduction announce-and-forget-rotten-fruits
:when (:detects (Rotten-Fruit ?f))
:perform (say-fruit-age ?f (forget-all-about ?f)))
```

That way, we are able to perform some basic inferences, using only information provided by the anchoring component, utilizing higher level reasoning. Hence, we can focus on the logic aspects and dynamic properties of objects, rather than computing sensory information directly. As an additional challenge for future work, using information about dynamic properties in the KB can be used to recognize again an object that has been previously perceived even if some of the properties have changed over time.

#### 4.4. 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 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 ...” followed by 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 its colour 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 routine, which collects candidates from the KB that

12 Amy Loutfi, Silvia Coradeschi, Marios Daoutis, Jonas Melchert

match the description. If there is more than one candidate, the anchoring module checks for further properties in the given description, apart from shape and colour, and selects additional properties. If an ambiguity still persists, the system tries to ask the user about which of the candidate objects to select.

## 5. Experiments

### 5.1. TestBed

We have constructed a reference implementation of a symbiotic system, as well as built a demonstrator environment in which the following case studies can be performed.

The basic building blocks of our testbed is inspired by the notion of a the concept of PEIS-Ecology, originally proposed by Saffiotti and Broxvall<sup>16</sup> which combines insights from the fields of ambient intelligence and autonomous robotics, to generate a new approach to the inclusion of robotic technology into smart environments. In this approach, advanced robotic functionalities are not achieved through the development of extremely advanced robots, but rather through the cooperation of many simple robotic components called PEIS. A PEIS can be as simple as a toaster and as complex as a humanoid robot. In general, we define a PEIS to be a set of inter-connected software components, called PEIS-components, residing in one physical entity. Each component may include links to sensors and actuators, as well as input and output ports that connect it to other components in the same or another PEIS. A PEIS-Ecology is then a collection of inter-connected PEIS, all embedded in the same physical environment.

As part of the testbed, we have built a physical facility, called the PEIS-home, which looks like a typical bachelor apartment of about  $25m^2$ . It consists of a living room, a bedroom and a small kitchen. The PEIS-Home is equipped with communication and computation infrastructure, and with a number of sensors. Fig 5 shows a few snapshots of the home.

In our ecology there is an iRobot's Magellan Pro indoor research robot called Pippi. In addition to the usual sensors, the robot is equipped with a CCD color camera and with a Cyranose 320<sup>TM</sup> electronic nose used to identify and discriminate between odours. The Cyranose 320<sup>TM</sup> electronic nose is a contained unit which relies on a 32-channel carbon-black polymer composite chemiresistor array for odour sampling. The sensors on Pippi are mounted in such a way that line of sight and line of "smell" are coordinated. This is done by placing the e-nose below the camera and odour samples are drawn from a uni-directional air-flow at the front of the robot, as shown in Fig 5. The detection of an odour is therefore maximized when the robot is directly facing the source.

On-board Pippi also runs an instance of the Thinking Cap, an architecture for autonomous robot control based on fuzzy logic<sup>17</sup>, and an instance of the player program<sup>15</sup>, which provides a low-level interface between the robot's sensors and actuators and the PEIS-Ecology's tuple-space.

Pippi responds to tuples providing commands and requests. These include tuples of type `Goal`, providing navigation goals, and of type `Smell`, providing smell commands. Pippi produces, among others, tuples that indicate the state of the navigation and the ol-



Fig. 5. (Left) A rough sketch of the PEIS-Home with a picture of the control deck. (Lower Left), the kitchen, and common room. (Right) A view of Pippi with the Cyranose 320 mounted underneath the camera.

factory classification results which are then updated in the anchors and KB respectively. A schematic overview of the system components used and their connection is given in Fig 6, many of the components are separate PEIS components.

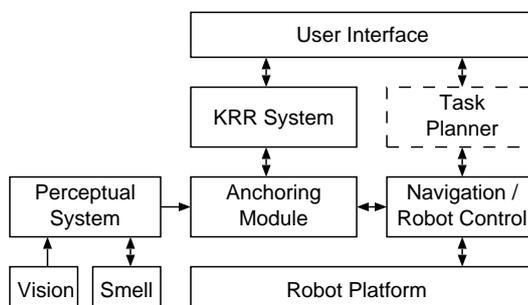


Fig. 6. Schematic overview of the system parts and their connections. The task planner is shown with dashed box as it is not directly to the work presented here.

### 5.2. Case-Studies

Three sets of experiments were performed. The aim of each experiment is twofold, first the examined case studies illustrate the ability of the KRR component to accept queries and return interpretable information (Human Robot Interaction). Secondly, they illustrate the ability of the KRR component to be used as an integral process of the anchoring module assisting in the resolution of ambiguities as discussed in Section 2 and recognizing relevant properties of objects.

14 Amy Loufi, Silvia Coradeschi, Marios Daoutis, Jonas Melchert

### 5.2.1. Spatial Relations for Finding Objects

In the first case, spatial relations are used to describe objects. Here the robot surveys a static scene with three objects (two green garbage cans and a red ball), See Fig 7, and the anchoring module creates anchors for these objects bottom-up (as soon as identified by the vision system). The output of the anchoring module is summarized as follows below (detailed description of the perceptual data is omitted):

```
(ANCHOR :NAME 'GAR-4 :ID 'ANCHOR-1 :SYMBOLIC-DESCRIPTION ' ((SHAPE=
GARBAGE) (COLOR= GREEN) :PERCEPTUAL-DESCRIPTION ...))

(ANCHOR :NAME 'BALL-2 :ID 'ANCHOR-2 :SYMBOLIC-DESCRIPTION ' ((SHAPE=
BALL) (COLOR= RED) :PERCEPTUAL-DESCRIPTION ...))

(ANCHOR :NAME 'GAR-5 :ID 'ANCHOR-3 :SYMBOLIC-DESCRIPTION ' ((SHAPE=
GARBAGE) (COLOR= GREEN) :PERCEPTUAL-DESCRIPTION ...))
```

The spatial relations for the objects are given in the following table and are stored in the KB.

Table 1. Spatial Relations for the anchored objects.

	GAR-4	BALL-2	GAR-5
GAR-4	—	(BEHIND 0.95)(LEFT 0.62)	(LEFT 0.94)
BALL-2	(FRONT 0.96) (RIGHT 0.2)	—	(FRONT 0.96) (LEFT 0.43)
GAR-5	(RIGHT 0.94)	(BEHIND 0.96) (RIGHT 0.85)	—

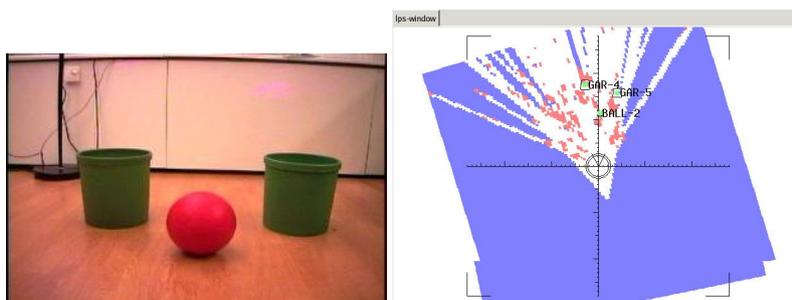


Fig. 7. Spatial Relations: scene from the robot's viewpoint (left) and snapshot of the robot's perceptual space with the created anchors (right).

It is now possible to use spatial relations to query for an object, for example: “find the green garbage left of ball”, returns ((ANCHOR ANCH-1 GAR-4 ...)) as result.

Similarly, a human user can be asked to resolve an ambiguity in a *find* request: in the scene from the previous example, the query is “Find the green garbage”. (This experiment is scripted and uses a simple pre-formulated scheme to guide the interaction with the user by text prompts.) As the find request returns more than one anchor (namely ANCH-1 and

ANCH-3), the script determines an anchored object that is spatially related to these anchors as reference object and presents the user with a choice, enumerating the returned anchors and their spatial relation(s) to the reference object. Then the query is reformulated using additionally the selected relation(s). For example:

```
? FIND THE GREEN GARBAGE
- FOUND 2 CANDIDATES: PLEASE CHOOSE
- 1. GREEN GARBAGE LEFT BEHIND OF RED BALL
- 2. GREEN GARBAGE RIGHT BEHIND OF RED BALL
? 1
- REFORMULATING...
- FOUND: ((ANCHOR ANCH-1 GAR-4 ...))
```

### 5.2.2. Coping with Multiple Modalities

In Section 4.1 we describe how simple inference can be used to deduce information about untraditional sensors, such as olfaction. In the flavour of the previous case described where an ambiguity is introduced, the robotic system uses its olfactory module for resolving an ambiguous case among different cups. Pippi is given the command “Find a green cup with coffee”. Four candidates are found, the KRR providing information that cups, which are vessels can contain liquid with an associated smell, triggers the task planner to generate a plan to visit each candidate and collect an odour property. The ambiguity is then resolved once a first match is found. An output of the camera image from the robot and the respective local perceptual space is given in Fig 8. The output of the anchoring module described below shows that at the perceptual signature of the object is updated with an odour property.

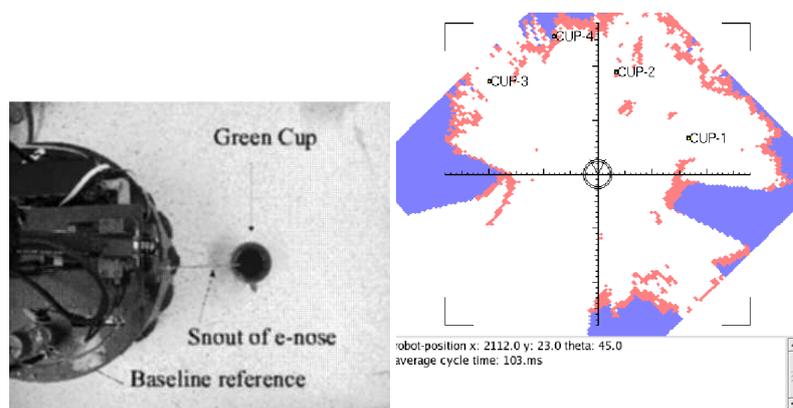


Fig. 8. KRR with olfaction: overhead view of the robot's as it approaches a cup to collect an odour sample (left) and snapshot of the robot's perceptual space with the created anchors showing all candidates (right).

```
(ANCHOR :NAME 'CUP-3 :ID 'ANCHOR-1 :SYMBOLIC-DESCRIPTION ' ((SHAPE=
CUP) (COLOR= GREEN) (SMELL = COFFEE))
:PERCEPTUAL-DESCRIPTION (TRAJECTORY :NAME 'CUP-3 :TIMESTAMP 85
:KNOXEL (KNOXEL :DOMAIN 'VISION :TIMESTAMP 85 :PROPERTIES ' ((SHAPE
CUP) (COLOR GREEN) (POSITION (1958.0 -347.0)))) :SMELL-DESCRIPTION (COFFEE) :LIFE 0.8)
```

16 Amy Loufi, Silvia Coradeschi, Marios Daoutis, Jonas Melchert

```
(ANCHOR :NAME 'CUP-4 :ID 'ANCHOR-2 :SYMBOLIC-DESCRIPTION '((SHAPE=
CUP) (COLOR= GREEN) (SMELL = ETHANOL))
:PERCEPTUAL-DESCRIPTION (TRAJECTORY :NAME 'CUP-4 :TIMESTAMP 85
:KNOXEL (KNOXEL :DOMAIN 'VISION :TIMESTAMP 85 :PROPERTIES '((SHAPE
CUP) (COLOR GREEN) (POSITION (2089.0 -154.0)))))) :SMELL-DESCRIPTION (ETHANOL) :LIFE 0.9)

(ANCHOR :NAME 'CUP-2 :ID 'ANCHOR-3 :SYMBOLIC-DESCRIPTION '((SHAPE=
CUP) (COLOR= GREEN))
:PERCEPTUAL-DESCRIPTION (TRAJECTORY :NAME 'CUP-2 :TIMESTAMP 85
:KNOXEL (KNOXEL :DOMAIN 'VISION :TIMESTAMP 85 :PROPERTIES '((SHAPE
CUP) (COLOR GREEN) (POSITION (2029.0 98.0)))))) :SMELL-DESCRIPTION NIL
:LIFE 1)

....
```

### 5.2.3. Managing Object Properties

Finally in section 4.3 concerning the objects' dynamic properties handled by the KB, we investigate the case where there are some scattered fruits on the floor, as show in Fig 9 (currently Pippi is too low to see objects on a table). From left to right we have one apple, one rotten banana, one fresh banana, and an orange. The robot navigates around the floor and correctly perceives and classifies the instances to the KB, obtaining the following anchors (LISP):

```
(ANCHOR :NAME 'APPLE-1 :ID 'ANCHOR-1 :SYMBOLIC-DESCRIPTION '((SHAPE
SPHERE)) :PERCEPTUAL-DESCRIPTION (TRAJECTORY :NAME 'APPLE-1 :TIMESTAMP 280
:KNOXEL (KNOXEL :DOMAIN 'VISION :TIMESTAMP 280 :PROPERTIES '((SHAPE
SPHERE) (COLOR RED) (POSITION (1100.0 -35.0)))))) :SMELL-DESCRIPTION NIL :LIFE 1)

(ANCHOR :NAME 'BANANA-1 :ID 'ANCHOR-2 :SYMBOLIC-DESCRIPTION '((SHAPE BANANA-SHAPE))
:PERCEPTUAL-DESCRIPTION (TRAJECTORY :NAME 'BANANA-1 :TIMESTAMP 280 :KNOXEL
(KNOXEL :DOMAIN 'VISION :TIMESTAMP 280 :PROPERTIES '((SHAPE BANANA-SHAPE) (COLOR
BLACK) (POSITION (1151.0 -5.0)))))) :SMELL-DESCRIPTION NIL :LIFE 1)

(ANCHOR :NAME 'BANANA-2 :ID 'ANCHOR-3 :SYMBOLIC-DESCRIPTION '((SHAPE
BANANA-SHAPE)) :PERCEPTUAL-DESCRIPTION (TRAJECTORY :NAME 'BANANA-2 :TIMESTAMP
280 :KNOXEL (KNOXEL :DOMAIN 'VISION :TIMESTAMP 280 :PROPERTIES '((SHAPE
BANANA-SHAPE) (COLOR YELLOW) (POSITION (1420.0 20.0)))))) :SMELL-DESCRIPTION
NIL :LIFE 1)

(ANCHOR :NAME 'ORANGE-1 :ID 'ANCHOR-4 :SYMBOLIC-DESCRIPTION '((SHAPE SPHERE))
:PERCEPTUAL-DESCRIPTION (TRAJECTORY :NAME 'ORANGE-1 :TIMESTAMP 280 :KNOXEL
(KNOXEL :DOMAIN 'VISION :TIMESTAMP 280 :PROPERTIES '((SHAPE BANANA-SHAPE) (COLOR
ORANE) (POSITION (1190.0 42.0)))))) :SMELL-DESCRIPTION NIL
:LIFE 1)
```

We then ask the robot to pick the orange that is left of the apple (spatial misinformation). Since this is not the case, it responds with a valid proposition, that there is no orange on the left of the apple, but there is one on the right of the banana. The user confirms that the alternative candidate is indeed the requested object.

```
? FIND THE ORANGE LEFT OF THE APPLE
- NO MATCH FOUND
- ALTERNATIVE CANDIDATE :
- 1. ORANGE RIGHT OF FRESH-BANANA
? 1
```

```
-REFORMULATING..
-FOUND: ((ANCHOR ANCHOR-4 ORANGE-1... ))
```

As a next step, we ask the robot to pick a banana on the right of the apple. Here, since there are 2 banana candidates right of the apple and one of them is being classified as rotten, it informs us about the situation and after deleting this option, it suggests that there is one banana on the right of the apple, (that is suitable for consumption) and returns this instance instead:

```
? FIND THE BANANA RIGHT OF APPLE
- FOUND 2 CANDIDATES :
- 1. ROTTEN-BANANA RIGHT OF APPLE
- 2. FRESH-BANANA RIGHT OF APPLE
- Fruit ROTTEN-BANANA is a rotten fruit!
- Instance Removed
-REFORMULATING ....
-FOUND: ((ANCHOR ANCHOR-3 BANANA-2... ))
```



Fig. 9. Several fruits as seen by the robot.

## 6. Related Work

There are only a handful of works that consider knowledge representation and reasoning for an anchoring framework per se. Primarily work involving formal knowledge representation for perceptual anchoring was done by Bonarini et al.<sup>1</sup> 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 organised 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.

Alternatively, Mastrogiovanni et al.<sup>14</sup> 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.

Integration of KRR for human robot interaction (HRI) has been studied in detail in a noteworthy approach that incorporates knowledge representation and inference mechanisms, to the ultimate goal of building a fully autonomous service/personal robot. Seabra Lopes et al.<sup>18,12</sup> describe in the project named C.A.R.L. ( Communication, Action, Reasoning and Learning in Robotics ) a way to utilize the KRR component for knowledge acquisition and information disambiguation. The KR language Carl uses is based on semantic networks and UML object diagrams. This approach promotes interoperability between the KRR module and the SLU (Spoken Language Understanding) module, both of which are equally vital to knowledge acquisition while making the robot capable of understanding concepts familiar to the human interlocutor. It is not mentioned however how the anchoring component is being implemented in their agent based approach.

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.<sup>10</sup> 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.

Also in cognitive vision systems, spatial reasoning has been studied in detail in some recent work presented by Kennedy et al.<sup>19</sup> that bring forth a cognitive architecture approach mostly implemented in ACT-R, to support spatial representation and reasoning. This formed the basis for the robot to interact with other team members to track and approach moving targets. Their spatial support layer is literally responsible for translating metric information to the cognitive map, which in turn performs some higher level reasoning about the position of the target. The StealthBot, reasoned in the cognitive level similarly to how people do, however it is considered mainly a domain specialized to the military operations and applications.

A more generalised approach by Kruijff et al.<sup>9</sup> investigates a spatio-temporal model for Human-Robot dialogue comprehension. They describe a combination of linguistic reasoning with reasoning about intentions and plans. Their focus is directed to the relation between spatiotemporal-causal aspects of linguistically conveyed meaning and the planning these aspects reflect. While they maintain a planning based approach, the referred Planner Memory (PM) somehow acts as an anchoring management system about actions, objects and time. Despite that most of the individual components have been implemented on the well-grounded theory, their current status indicates that the major integration of those components is taking place, into a system for collaborative human-robot interaction.

## 7. Future Work and Conclusions

In this work, we have used a KRR component to enhance an anchoring process in a symbiotic robotic system. In the future work we aim 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 use of DOLCE's Quale concept for property values has not been addressed, but will be part of future work. The linguistic HRI part is still very rudimental and based on a text interface and reminds of the capabilities of Winograd's SHRDLU system<sup>20</sup>. We intend to use a simple speech dialogue system in future work, similar to the system of Hois et al.<sup>10</sup>. 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<sup>6</sup> 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. Still, however, our work is the first to use a KRR system for the anchoring process. This 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 facilitates the management of information and queries on the anchored objects can take a more advanced form. The given examples, 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 the context of symbiotic systems where robot, human and environment cooperate.

**Acknowledgements:** This work has been supported by the Swedish Research Council and the Swedish KK foundation.

## References

1. Andrea Bonarini, Matteo Matteucci, and Marcello Restelli. Concepts for anchoring in robots. In *Proceedings of AI\*IA 2002, 7th National Congress of the Italian Association for Artificial Intelligence*, Bari, Italy, 2002.
2. Dave Brill. Loom reference manual, for Loom version 2.0. Technical report, ISI, University of Southern California, USA, 1993.
3. M. Broxvall, S. Coradeschi, L. Karlsson, and A. Saffiotti. Recovery planning for ambiguous cases in perceptual anchoring. In *Proceedings of AAAI-05, 20th National Conference on Artificial Intelligence*, Pittsburgh, USA, 2005.
4. S. Coradeschi and A. Saffiotti. Anchoring symbols to sensor data: Preliminary report. In *Proceedings of AAAI-2000, 17th National Conference on Artificial Intelligence*, Austin, USA, 2000.
5. S. Coradeschi and A. Saffiotti. Symbiotic robotic systems: Humans, robots, and smart environments. *IEEE Intelligent Systems*, 21(3):82–84, 2006.
6. T. Fong, I. Nourbakhsh, R. Ambrose, R. Simmons, A. Schultz, and J. Scholtz. The peer-to-peer human-robot interaction project. In *Proceedings of AIAA Space 2005 Conference*, Long Beach, CA, USA, 2005.
7. 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.

20 Amy Loutfi, Silvia Coradeschi, Marios Daoutis, Jonas Melchert

8. Peter Gärdenfors. *Conceptual Spaces: The Geometry of Thought*. MIT Press, 2000.
9. Michael Brenner Geert-Jan M. Kruijff. Modelling spatio-temporal comprehension in situated human-robot dialogue as reasoning about intentions and plans. In *Symposium on Intentions in Intelligent Systems, AAAI Spring Symposium Series 2007, Stanford University, Palo Alto, CA, USA, 2007*.
10. J. Hois, M. Wünstel, J.A. Bateman, and T. Röfer. Dialog-based 3D-image recognition using a domain ontology. In *Proceedings of the International Conference Spatial Cognition 2006, Bremen, Germany, 2006*.
11. A. Loutfi, S. Coradeschi, and A. Saffiotti. Maintaining coherent perceptual information using anchoring. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05)*, Edinburgh, Scotland, 2005.
12. Marcelo Quindere Mario Rodrigues Luis Seabra Lopes, Antonio J.S. Teixeira. From robust spoken language understanding to knowledge acquisition and management. In *Proceedings of Interspeech 2005, Lisboa, Portugal, p. 3469-3472., 2005*.
13. Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, Alessandro Oltramari, and Luc 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, 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, 2007*.
15. Player/Stage Project. [playerstage.sourceforge.net/](http://playerstage.sourceforge.net/).
16. A. Saffiotti and M. Broxvall. PEIS ecologies: Ambient intelligence meets autonomous robotics. In *Proc of the Int Conf on Smart Objects and Ambient Intelligence (sOc-EUSAI)*, pages 275–280, Grenoble, France, 2005.
17. A. Saffiotti, K. Konolige, and E. H. Ruspini. A multivalued-logic approach to integrating planning and control. *Artificial Intelligence*, 76(1-2):481–526, 1995.
18. Antonio Teixeira. Spoken language interface for intelligent service robots. Microsoft Workshop on Speech Technology, Microsoft Portugal, Taguspark, Porto Salvo, May 2007.
19. Matthew Marge William Adams Benjamin R. Fransen Dennis Perzanowski Alan C. Schultz William G. Kennedy, Magdalena D. Bugajska and J. Gregory Trafton. From spatial representation and reasoning for human-robot collaboration. In *Proceedings of the Twenty-Second Conference on Artificial intelligence. Vancouver, Canada (pg 1554-1559), 2007*.
20. Terry Winograd. Procedures as a representation for data in a computer program for understanding natural language. Technical Report AITR-235, MIT Artificial Intelligence Laboratory, Boston, USA, 1971.