Cooperative Knowledge Based Perceptual Anchoring

Marios Daoutis\textsuperscript{(a)} and Silvia Coradeschi and Amy Loutfi

Cognitive Robotics Lab, Center for Applied Autonomous Sensor Systems Research, Dept. of Science and Technology
Örebro University, SE-70182 Örebro, Sweden

Received July 9, 2010
Revised December 3, 2011
Accepted (Day Month Year)

Abstract: In settings where heterogenous robotic systems interact with humans, information from the environment must be systematically captured, organized and maintained in time. In this work, we propose a model for connecting perceptual information to semantic information in a multi-agent setting. In particular, we present semantic cooperative perceptual anchoring, that captures collectively acquired perceptual information and connects it to semantically expressed commonsense knowledge. We describe how we implemented the proposed model in a smart environment, using different modern perceptual and knowledge representation techniques. We present the results of the system and investigate different scenarios in which we use the commonsense together with perceptual knowledge, for communication, reasoning and exchange of information.

Keywords: cognitive robotics; physical symbol grounding; commonsense information; multi-agent perception; object recognition

1. Introduction

In this work we explore the problem of connecting perceptions of physical objects, with the corresponding conceptual knowledge and common-sense information, in the context of cognitive agents (i.e. mobile robots, smart environments). We are primarily motivated by the need to design a robot that monitors, infers, plans, assists and communicates in a household environment in the presence of humans. Consider a future scenario where an elderly person communicates with their personal robotic assistant about different objects (and their functions, locations) that are present in the environment. A example of a query involving objects could be “Where is my medication” or “where are my keys”. To handle such queries, a mechanism that establishes the link between perception and conceptual knowledge is needed. In the literature this is refered as the “Symbol Grounding” (SG)\textsuperscript{16} and describes the

\footnotesize
\textsuperscript{a} (a) Marios Daoutis \cdot Cognitive Robotics Lab.
A.A.S.S \cdot Dept. of Science and Technology, Örebro University,
SE-70182 Örebro, Sweden \cdot e-mail: marios.daoutis@oru.se
establishment of the perceptual-symbolic correspondences in cognitive agents, and the attribution of meaning to the abstract concepts.

1.1. Symbol Grounding in Robotics

In robotics, the SG problem is approached from quite diverse points of view which briefly concern linguistic, social, and technical aspects with respect to attributing meaning to concepts. The Physical Symbol Grounding Problem according to Vogt states that by using a semiotic definition\(^a\) of symbols, the SG problem is transformed into the technical problem of constructing the triadic relation between referent, meaning and form\(^4\), since the semiotic symbols are by definition meaningful and grounded\(^2\). He approaches the physical symbol grounding problem from the perspective of embodied robotics\(^1,2\) where the agent interacts with the environment, grounding its symbolic system in the sensorimotor activations. In a multi-robot system, the problem becomes more complex as symbols not only are grounded, but commonly shared. This is called the “social grounding problem”\(^4\). Finally in the context of applied robotics, according to Coradeschi and Saffiotti\(^7\), the link between the agent’s low level sensing processes (i.e. vision, localization) and the high level semantic representations is called Perceptual Anchoring (See Fig. 1\(^b\)). It is important to highlight the differences of perceptual anchoring with physical symbol grounding. Anchoring can be seen as a special case of symbol grounding, which concerns primarily physical objects\(^4\), actions\(^8\) and plans\(^6\) but not abstract concepts\(^17\). Even

\(^a\)According to a semiotic definition, symbols have (1) a form (Peirce’s “representamen”), which is the physical shape taken by the actual sign; (2) a meaning (Peirce’s “interpretant”), which is the semantic content of the sign; and (3) a referent (Peirce’s “object”), which is the object to which the sign refers. Following this Peircean definition, a symbol always comprises a form, a meaning and a referent, with the meaning arising from a functional relation between the form and the referent, through the process of semiosis or interpretation.

\(^b\)We should note here that in this particular approach we focus explicitly on Vision and Localization from the sensing modalities, and Communication and Inference from the cognitive abilities.
though anchoring elaborates on certain aspects of physical symbol grounding, it is fundamentally another solution, as it deviates from the scope of language development (according to Vogt). In this context, anchoring attempts to provide a model for grounding symbols to perceptual data from a physical object. For example, in symbol grounding we would consider ways to generally ground the concept of “stripes” or “redness” in perception, while in anchoring we would instead consider ways of modelling the perceptual experiences of a cognitive robot into a structure that a) describes an object, its properties and attributes (both symbolically and perceptually) as well as b) maintain in time these descriptions.

1.2. Perceptual Anchoring

Since it was first introduced by Coradeschi and Saffiotti works in anchoring range from single robot scenarios to multi-robot *Cooperative Perceptual Anchoring*, and from multiple modalities such as vision and olfaction to the integration of non-trivial complex symbolic systems including common-sense information. With the term *cooperative anchoring* we refer to the case where perceptual information is distributed across multiple agents. Work done by LeBlanc and Saffiotti formalizes and implements a cooperative anchoring framework which computes and distributes perceptual information globally based on local individual perceptions which are transferred between robots. Our previous work has explored the integration of high-level *conceptual knowledge* on a single agent, via the combination of a fully fledged Knowledge Representation and Reasoning (KR&R) system with the anchoring framework. More specifically, we have examined the use of semantic knowledge and common-sense information so as to enable reasoning about the perceived objects at the *conceptual level*.

In the present work, we consider cooperative anchoring scenarios in which *semantic information* is distributed and exchanged across multiple agents. We use perceptual anchoring for managing perceptually grounded knowledge. We build further on the ideas introduced in our previous KR&R approaches while borrowing concepts from the cooperative anchoring framework proposed by LeBlanc and Saffiotti by adding support for semantic information. This enables the different robotic systems to work in a cooperative manner acquiring perceptual information and contributing to the semantic collective knowledge base (KB). Our work addresses the distributed anchoring problem, similarly adopting *local* and *global* anchor spaces, but a *global anchoring space* is defined in a *centralized* way instead of a distributed one. The different perceptual agents maintain their local anchoring spaces, while the global anchoring space is coordinated by a centralized global anchoring component which hosts the knowledge representation system. The reasons behind the centralized module is the computational complexity of the common-sense knowledge base and the difficulty to model distributed knowledge representation. We show how this integration, which relies on common-sense information, is capable of supporting better (closer to linguistic) human robot communication and improves
the knowledge and reasoning abilities of an agent. This focus of this paper is (1) the theoretical framework of the perception and symbol grounding of the different anchoring agents (2) the integration with a large scale deterministic common-sense knowledge base and (3) details about the implementation and instantiation of this framework in a smart home-like environment containing heterogeneous robotic agents.

In Section 2 we introduce the anchoring computational framework which defines the local/global layers and the different functionalities to transverse through the anchoring spaces. Then in § 3 we introduce the knowledge representation techniques used to design and implement the framework. In § 4 we present the test-bed used in the evaluation with details on the perceptual modalities and symbol grounding techniques used in the implementation. In § 5 we give details on the performance of the implemented system as well as we present the different scenarios. Finally in § 6 we present an overview of related approaches while in § 7 we summarize the paper.

2. The Anchoring Framework

The main task of anchoring is to systematically create and maintain in time the correspondences between symbols and sensor data that refer to the same physical object. Given a symbol system and a perceptual system, an anchor \( \alpha(t) \), indexed by time, is the data structure that links the two systems. In this work, Perceptual Anchoring Management is the process which produces and updates anchors for the existence of every physical object in the environment. The symbol system refers to the collection of all abstract hierarchically structured symbolic knowledge, expressed in some form of formal logic, containing the set of concepts, instances, and sets of relations and rules between the concepts. The perceptual system, is mainly considered as a group of sensors and feature extraction algorithms (which process sensor data), that are available in a perceptual agent and continuously produce percepts and attributes which are structured collections of measurements assumed to originate from the same physical object. An attribute is a measurable property of a percept. The set of attribute-value pairs of a percept is called the perceptual signature. Furthermore, a perceptual agent is conceived as a mobile robot, or the ambient perceptual agent of an intelligent home, or more generally (possibly heterogeneous) groups of sensors and actuators which operate in an environment.

2.1. Computational Framework Description

We consider some environment, \( \mathcal{E} \), which contains \( n \) perceptual agents or robots \( \{Ag_1, Ag_2, \ldots, Ag_n\} \), where \( n > 0 \). Each perceptual agent \( Ag_i \), has a perceptual system \( S(Ag_i) = \{S(Ag_i)^1, S(Ag_i)^2, \ldots, S(Ag_i)^k\} \), \( k > 0 \), composed of sensors \( S(Ag_i)^k \) which produce perceptual data either in raw sensor measurements or via feature extraction, feature sets on a domain or feature space \( D(S(Ag_i)^k) \). For example, a camera that generates series of images which are then processed by a feature extraction algorithm (e.g. SIFT) to produce a vector of features in that domain. Similarly,
a laser range-finder generates a vector of points (or features). So we call $D(Agi)$ the perceptual space of an agent, the set of all domains $D(Agi) = \bigcup_k D(S(Agi))^k$ for every sensor in $S(Agi)$. At any time instant $t$, the perceptual signature of the anchor $\pi(x,t) = \{\pi_1(x,t), \pi_2(x,t), \ldots\}$ is the collection of percepts which are assumed to originate from the one physical object $x$. A percept $\pi(x,t)$ is a partial observation at time $t$, represented by a mapping from the sensor $S^k$ of the agent $Agi$ to the perceptual domain $D$ for that sensor:

$$\pi(x,t) : S(Agi)^k \mapsto D(S(Agi))^k$$  \hspace{1cm} (1)

The next step in the anchoring process, is the predicate grounding. During the predicate grounding in each perceptual agent, the collection of percepts of each perceptual signature is grounded to unary predicates via a predicate grounding relation $g$, which essentially maps perceptual attributes to predicate symbols from the set $\mathcal{P} = \{p_1, p_2, \ldots\}$ of predicate symbols.

$$g_{\text{local}}^\text{modality} \subseteq \mathcal{P} \times \pi(x,t)$$  \hspace{1cm} (2)

As for example consider that a mobile robot $Agi$ has a camera sensor $S(Agi)^1$ that produces images. A feature extraction technique extracts colour percepts $\pi_1(x)$ from the images that correspond to the domain of the
HSL colour space $\mathcal{D}(S(Ag_1)^1) = \{H,S,L\} = \{[0,360),[0,1],[0,1]\}$ so that at time $t$, $\pi_1(x,t) : S(Ag_1)^1(x,t) \rightarrow \mathcal{D}(S(Ag_1)^1) = (0^\circ,1,0,5)$ indicates that the percept $\pi_1(x)$ corresponds to the red colour of $x$ at time $t$. During the predicate grounding a classification technique processes the extracted features and grounds the percept $\pi_1(x)$ with the predicate ‘red’, eg : the $g_{\text{colour}} \rightarrow \text{‘red’} \times S(Ag_1)^1 \rightarrow (0^\circ,1,0,5))$. Similarly there might also exist other grounding relations for different modalities, such as $g_{\text{size}}$ with predicate symbol set $p_{\text{size}} = \{p_{\text{size}}(1) : \text{‘small’}, p_{\text{size}}(2) : \text{‘large’}, \ldots\}$, or $g_{\text{object}}$ with predicate symbol set $p_{\text{object}} = \{p_{\text{object}}(1) : \text{‘book’}, p_{\text{object}}(2) : \text{‘cup’}, \ldots\}$. At this point we can say that the symbolic description is the set $\sigma(\pi(x))$ of all grounded unary predicates from the perceptual signature of an anchor. For example a symbolic description about a small sized red cup, would be formed in the following way: $\sigma(\pi_1(x,t)) = \{p_{\text{size}}\text{‘small’}, p_{\text{object}}\text{‘cup’}, p_{\text{colour}}\text{‘red’}\}$.

We consider a final element, which is the knowledge fragment which corresponds to the grounded symbolic predicates from the previous step. As seen from Fig. 2 during this step essentially we extract a fragment of the hierarchically structured knowledge which includes the concepts and relations that regard the grounded symbolic predicates that correspond to percepts originating from the same physical object. Therefore we have a knowledge representation system $\mathbb{KB}$ which contains: an ontology, concepts and their relations in the domain of common-sense knowledge and a reasoning engine that operates on this knowledge base. During knowledge extraction, we query for every predicate that belongs to the symbolic description (e.g. $\sigma = \{p_{\text{local}}\text{‘small’}, p_{\text{object}}\text{‘cup’}, p_{\text{colour}}\text{‘red’}\}$), and we extract the fragment of the ontology that contains all the concepts and their direct relations and generalizations that represent the modality and the grounded information into $\omega(\sigma) \in \mathbb{KB}$. We consider a knowledge translation mechanism, which translates the set of the grounded predicates into the concepts from the extracted ontology fragment. An example is shown below in Fig. 3.

$$k \subseteq \mathbb{KB} \times \sigma(\pi(x,t)) = \omega(\sigma(\pi))$$ (3)

So at every moment $t$, one anchor $\alpha(x,t)$ of an object $x$, contains four elements: the perceptual signature $\pi$, the symbolic description $\sigma$, the knowledge fragment $\omega$ of the corresponding object, and the unique identifier meant to denote this anchor in the anchoring space. The anchor $\alpha$ is grounded at time $t$, if it contains the percepts perceived at $t$ and the corresponding descriptions and knowledge. If the object is not observable at $t$ and so the anchor is ungrounded, then no percept is stored in the anchor but the description as the anchor remains in the anchoring space, to provide the best available estimate since the last observation (i.e. memory).

$$\alpha(x,t) = \{\text{uuid}, \pi(x,t), \sigma, \omega\}, \text{ where}$$

$$\pi(x,t) = \begin{cases} \emptyset & \text{if } \exists \pi(x,t) : S(Ag_i,t)^k \rightarrow \mathcal{D}(S(Ag_i)^k) \\ \pi(x,t) & \text{if } \exists \pi(x,t) : S(Ag_i,t)^k \rightarrow \mathcal{D}(S(Ag_i)^k) \end{cases}$$ (4)
Percepts are grounded to predicates using (2) and predicates are translated to knowledge using (3). We can then represent the local anchoring space of an agent, by $A(A_g)^{local} = \{ \alpha(x,t), \alpha(y,t), \alpha \ldots \}$. It is essentially a multidimensional space where heterogeneous items of information are mapped and represents the current perceptual and conceptual state of an agent, described in terms of the perceptual signatures, their symbolic descriptions and the corresponding knowledge tangible to the symbolic descriptions (see Fig. 3).

2.2. Global Perceptual Anchoring

So far we have described the basic components of anchoring at the agent or local level. Similarly there is global anchoring management which supervises the local anchoring processes of each agent, and mediates the knowledge between the higher level knowledge representation and reasoning engine and the different anchoring agents. Once the correspondences of the local anchors has been established we unify the information into a global anchoring space with each anchor representing one physical object, that might have been perceived by one or many different perceptual agents. Let us consider one global anchoring space $A^{global}$ in which we unify the different agents’ local anchoring spaces, then we perform complementary symbol grounding on the global anchor information, while finally we augment the knowledge gathered from the global anchoring space into the knowledge representation system (KB).

First we unify the different individual anchoring spaces $A(A_g)^{local}$ into the $A^{global}$. This is assisted by the match functionality which is explained further in § 2.3. As we also see in Fig. 4 during this stage, the global grounding function takes as input multiple percepts $\pi_i(x,t)$ across different agents for the computation.
of global spatial/topological relations for example. Similarly we have the global grounding relations:

$$g_{\text{modality}}^{\text{Global}} \subseteq P_{\text{Global}} \times \bigcup_{\sigma} \pi_1(x,t)$$  \hspace{1cm} (6)

Like before, the grounding of spatial relations requires the position information from other objects (or anchors) \(g_{\text{SpatialRel}}^{\text{Global}}\) which further uses a symbol set \(P_{\text{SpatialRel}} = \{p_{1}^{\text{SpatialRel}}, 'leftOf', p_{2}^{\text{SpatialRel}}, 'near', \ldots\}\). Therefore we complement the global anchors’ grounded knowledge acquired by the different perceptual agents using the global grounding relations. The next step is to attach the knowledge fragment that corresponds to the newly grounded concepts by using (3) enriching the ontology fragment \(\omega(\sigma)\) of each global anchor. One might consider that the global anchoring space \(A^{\text{Global}}\) represents the collective perceptual memory of all agents. The final step before reaching the knowledge base, concerns the knowledge augmentation and synchronization of the symbolic knowledge of the global anchoring space into the
through a translation function:

\[ K^{Global} : \bigcup_{0}^{N} \omega(\sigma) \mapsto \mathbb{KB}, \text{where } \sigma \in A^{Global} \]  

(7)

Via the perceptual anchoring process we have described, how perceptual information from the environment \( E \), is mediated systematically into the \( \mathbb{KB} \) from one or more agents.

2.3. Anchor Management

To support the proposed framework we further describe the different functionalities across all the anchoring framework which are enabling the support of both top down and bottom up approaches. Essentially the terms “top down” and “bottom up” represent the two ways of traversing within the anchoring space, symbolically and perceptually respectively. “Bottom up” information acquisition is driven by an event originating from a sensing resource (e.g. the recognition of an object in an image) as soon as there are new percepts generated in the perceptual system in a continuous manner. This leads to the creation of an anchor and finally the knowledge regarding this anchor. While in “top down” acquisition, a symbolic concept or proposition from the knowledge representation system is anchored to percepts on request (e.g. from a planner). This is done by traversing downward into the anchoring space and where possible, converge into an anchor that contains percepts and symbols compatible with the specified concept/proposition. We therefore need different functionalities to be able to utilize this perceptual information management. Namely these are **(Re)Acquire**, **Track** and **Augment** functionalities for bottom up acquisition and for top down retrieval we have the **Query** and **Find** functionalities as they are depicted in Fig. 5.

2.3.1. Bottom-up Information Acquisition

**(Re)Acquire** Initiates a new anchor whenever a percept(s) is received which currently does not match any existing anchor. It takes a percept(s) \( \pi \), and returns an anchor \( \alpha \) defined at \( t \) and undefined elsewhere. To make this problem tractable, a priori information is given with regard to which percepts to consider. In bottom-up acquisition, a randomly generated symbol (Unique Identifier) is attributed to the anchor. Furthermore, information about the object and its properties and relations are included into the anchor via the knowledge translation mechanism described in § 2.1, so as to enable local symbolic reasoning.

**Track** Takes an anchor \( \alpha \) defined for \( t - k \) and extends its definition to \( t \). The track assures that the percept of the anchor is the most recent and adequate perceptual representation of the object. We consider that the signatures can be updated as well as replaced but by preserving the anchor structure over time we affirm the persistence of the object so that it can be used even when the object
is not currently perceived (caused by the object being out of view and/or by the errors generated during the measurement of perceptual data). This facilitates the maintenance of information while the agent/robot is moving. It also supports stable representations of the world at a symbolic level, in the long term and without carrying perceptual glitches.

**Augment** At each perceptual cycle, this function loops through the symbolic descriptions of the anchoring space and by keeping track of changes in the anchors, it appends the changes into the Knowledge Base. It can be seen as a “symbolic track” which essentially keeps the perceptual knowledge expressed in the knowledge representation system consistent with the symbolic descriptions which are grounded using the perceptual signatures of the observed objects in the environment. While at first this function appears trivial, synchronizing the concepts, instances and relations of the objects in the anchoring space with the ones in the knowledge base, regarding multiple perceptual agents, requires further verification so as to avoid conflicts or unrelated assertions. A motivating example is a cooperative symbolic anchoring scenario where the following incidents take place: An *Agent A* perceives that “a book is located on the table” and another *Agent B* perceives “the same book that is located on the table”. First during *Agent A*’s Augment the concepts ‘Book_1’ and ‘Table_1’ are created. Then when *Agent B* reaches augment, his perception of the book and the table must point to the same ‘Book_1’ and ‘Table_1’ concepts that were created by *Agent A* in the central common knowledge base. Hence in the **augment** function we account for such situations using information from the unified...
global anchoring space.

2.3.2. Top-down Information Acquisition

**Find** Takes a symbol or a symbolic description and returns an anchor $\alpha$ defined at $t$. It checks if existing anchors that have already been created by the (Re)Acquire satisfy the symbolic description, and in that case, it returns the matches. Otherwise, it performs a similar check with existing percepts (in case, the description does not satisfy the constraint of percepts considered by the (Re)Acquire). If a matching percept is found an anchor is retrieved. Anchor matching can be either partial or complete. It is partial if all the observed properties in the percept or anchor match the description, but there are some properties in the description that have not been observed whereas complete if all the observed properties in the percept or anchor match the description.

**Query** Accepts a logical formula originating from an external component such as a planner or a natural language interface, that needs to be inferred from perceptual observations and utilizes the anchoring capabilities and knowledge store (knowledge fragments) to produce the satisfiability of the formula. Query can be understood better if we describe it as the semantic “Find”. While “Find”, accepts a symbolic description and converges into anchors that contained the compatible percepts, “Query” accepts a logical formula, and converges into grounded concepts from the anchors in the anchoring space. In a concrete scenario, this logical formula could be a precondition axiom of a planner that needs to be satisfied in order for a plan to be executed. For a plan to fetch the book from the kitchen an axiom could be that a book exists in the kitchen $\exists(x, y) : (\text{Book}(x) \land \text{Kitchen}(y)) \land \text{ObjectFoundInLocation}(x, y)$. A similar query could have been generated from a natural language interface that processed a command to find a book in the kitchen. This formula via the query functionality is addressed to the perceptual anchoring agent, which in turn using its own symbolic reasoning capabilities and perceptual knowledge can infer, that it perceives an anchor which is bound to the concept ‘Book_1’ which is an instance of the concept ‘Book’ and is located in the instance ‘Kitchen_1’ which is an instance of the concept ‘Kitchen’ and hence satisfies the query for the bindings ‘Book_1’ and ‘Kitchen_1’ belonging to the anchor about the book in the kitchen.

2.3.3. Match Function

**Match** Is a multi-modal function which may accept percepts or symbols or both, and returns the anchors which match the required specification(s), by matching all the symbols (if any) against the symbolic descriptions of the anchoring space, and by matching all the percepts against the perceptual signatures of the (symbolically) matched anchors.
3. Knowledge Representation in Anchoring

As we see in § 2, the anchoring process is tightly interconnected with a knowledge representation system, which mainly contains common-sense knowledge about the world. It provides an abstract semantic layer that forms the base for establishing a common language and semantics (i.e. ontology) to be used among robotic agents and humans. While the development of sophisticated common-sense knowledge bases is an active research topic with interesting results (Open Mind Common Sense\textsuperscript{39}, WordNet\textsuperscript{12} or Cyc\textsuperscript{25}), we find that there is not much attention on how to integrate them, with grounded perception in mobile robots (mainly concerning real world applications and complexity). Our motivation behind the use of a common-sense knowledge base on a real robotic system, lies in our belief that future trends would want the knowledge bases and knowledge representation systems to move from simplistic and domain specific designs to large scale information (including common-sense) processing and reasoning engines which continuously acquire new knowledge from the web to enhance the reasoning capabilities of the robot or intelligent agents. Advances in recognizing textual entailment and semantic parsing, or in the expressiveness of modern formal logic languages, show that constructing such engines is closer to realization. Some examples include the NELL system (Never-Ending Language Learner) that extracts structured information from unstructured web pages\textsuperscript{3} or the True Knowledge semantic search engine\textsuperscript{c} that gives direct semantically evaluated answers to questions, using Natural Language.

3.1. 

Cyc Knowledge Base
The common-sense knowledge base we adopt in this work as the main building block of the knowledge representation, is Cyc\textsuperscript{25}, which captures an upper ontology, a knowledge base of everyday common sense knowledge and an inference engine, with the goal of enabling AI applications to perform human-like reasoning. The KB contains over three million human-defined assertions about 250,000 concepts, rules and common sense ideas. These are formulated in the language CycL, which is an extension of first-order predicate calculus with higher-order extensions\textsuperscript{d} and has a syntax similar to that of the Lisp programming language. The extensions mainly include features of higher order logics (e.g. quantification over predicates, functions, and sentences) that allow more expressive assertions. The role of Cyc is to maintain a world model which consists of the collection of the semantic information perceived by the different agents. In addition to the concepts and relations which capture common-sense knowledge, we must be able to express into the KB, knowledge about: (a) the agent itself (e.g. its entity description, its abilities etc.), (b) the perceptual attributes and their properties, (c) the concepts, relations and rules about the perceived objects and (d) the concepts that are going to be used

\textsuperscript{3} http://www.trueknowledge.com
\textsuperscript{c}expressive, unambiguous and enables automated reasoning
Cooperative Knowledge Based Perceptual Anchoring

3.2. Ontology

An ontology is an explicit specification of a conceptualization that involves concepts and their relations in a domain or a shared understanding of some domain. It is a hierarchically structured formally expressed piece of knowledge and an integral part of every intelligent agent. It mainly allows the agent to make sense of its knowledge about the environment, and forms the common notions and shared vocabulary between agents or an agent and a human. An upper (or top-level) ontology is an ontology which describes very general concepts that are shared across all knowledge domains and the most important aspect is to support very broad semantic interoperability between many other ontologies accessible “under” this upper ontology. The Cyc ontology adheres to the design criteria for ontologies whose purpose is knowledge sharing and interoperability among programs based on a shared conceptualization, proposed by Gruber\textsuperscript{15}, which promote clarity, coherence, monotonic extensibility, minimal encoding bias and minimal ontological commitment. Here monotonic extensibility means that new general or specialized terms can be included in the ontology in such a way that it does not require the revision of existing definition while clarity is satisfied when the structure of terms implies the separation between non-similar terms.

We adopt the Cyc upper ontology as one common ontology to be used in our implementation so as to enhance interoperability, exchange and reuse of knowledge. We have investigated also other candidates to be used as the foundation ontology such as WordNet\textsuperscript{12}, which was originally designed as a semantic network and qualifies as an upper ontology, by including the most general concepts, but is not formally axiomatized so as to make the logical relations between the concepts precise. DOLCE + DnS (A Descriptive Ontology for Linguistic and Cognitive Engineering)\textsuperscript{29}, have a clear cognitive bias, trying to capture the ontological categories underlying natural language and common-sense, by introducing categories as cognitive artifacts, which are dependent on human perception. However they are particularly devoted to the treatment of social entities, such as organizations, collectives, plans, norms, information objects but most importantly they require significant effort in manually modelling, capturing and populating the domain of discourse from scratch if that domain is in the context of common-sense information.

Besides the substantial amount of content distributed with the Cyc ontology, we should mention that there are great efforts to map the Cyc concepts, with Wikipedia concepts\textsuperscript{31}, with WordNet\textsuperscript{34}, with DBpedia\textsuperscript{34}, UMBEL\textsuperscript{e} and many more. This is of importance, because as the different objects are grounded in an ontology with connections to the semantic web, we allow semantic web technologies to access the

*(http://umbel.org/resources/about)*
knowledge of the robot and also be accessed by the robot. For instance (as we also see in the experimental scenarios § 5.2.4) when a robot needs to acquire a new concept, it can query about this concept the semantic web, and then complement its ontology with the newly fetched knowledge about the concept. Finally here we should mention that Cyc is conceptually divided into the upper ontology and several domain-specific ontologies (called micro-theories - Mts) which represent different contexts and are collections of concepts and facts pertaining to one particular realm of knowledge. For example micro-theories could be about space, time, people, perception, etc. Each Mt is individually not allowing any contradictions, however, contradictions are allowed between different Mts.

3.3. Semantic Integration Overview

Semantic integration takes place at two levels, in a similar way as the anchoring framework is organized. We have a local knowledge base which is part of each anchoring process of every agent. Each local KB is a subset of a global KB that is part of the global anchoring agent which manages the different local anchoring processes. The duality of the knowledge representation systems, is due to the fact, that we want to enable semantic knowledge and reasoning capabilities on each perceptual agent, but at the same time we cannot afford to embed a complete common-sense knowledge base instance on each agent. The result would either mean a misuse of computational resources, or that each perceptual agent, such as a mobile robot, does not have the required processing power to support a heavy knowledge base instance. On the other hand, we would like to inherit common-sense knowledge fragments from the superset of common-sense knowledge represented in the global common-sense knowledge base, on every agent. So our solution involves a local knowledge representation layer which is aiming at a higher degree of autonomy and portability and must: (a) be a lightweight component, (b) adhere to the formally defined semantics of the global knowledge base, (c) provide simple and fast inference, (d) allow communication between other components (e.g. the global anchoring agent, Cyc or the Semantic Web) and (e) be based on the same principles that our core knowledge representation system (Cyc) is based. While the global knowledge representation should be used as the central conceptual database and should: (a) define the context of perception within the scope the perceptual agents, (b) be based on formally defined semantics, (c) provide more complex reasoning, (d) contain the full spectrum of knowledge of the KB, (e) allow communication between other components (e.g. the local anchoring agents, Semantic Web) and (f) provide the fundamental knowledge principles that the local agents will use.

3.3.1. Local Semantic Anchoring Layer

Our approach for the local semantic layer is based on OWL, a family of knowledge representation languages for authoring ontologies and it is endorsed by the

\[\text{http://www.w3.org/TR/owl-features/}\]
World Wide Web Consortium\textsuperscript{8}. The semantics of OWL correspond to a translation of Description Logic. Therefore OWL has both syntax for describing ontologies and also formally defined semantics that gives meaning to the syntactic structures. To allow communication and knowledge exchange, between the different components but also between agents (and eventually the semantic web), the format in which OWL fragments are expressed is based on the Resource Description Framework (RDF)\textsuperscript{b} in XML format. RDF represents data as triples; subject, predicate and its value. RDF Schema extends this binary propositional expressivity to that of a semantic network. To enable inference, OWL/RDF entities exist within Jena\textsuperscript{i}, an open source Semantic Web and Reasoning Framework which provides built-in OWL reasoning and rule inference support, where queries can be handled in SPARQL\textsuperscript{j} format (RDF query language and a key semantic web technology)\textsuperscript{38}. The knowledge base of every perceptual agent consists of two parts: a terminological component $T$, called T-Box; and an assertional component $A$, called A-Box. In general, the T-Box contains sentences describing concept hierarchies (i.e. relations between concepts) while the A-Box contains grounded sentences stating where in the hierarchy individuals belong (i.e. relations between individuals and concepts). We can then define the knowledge base of an agent as $\mathbb{KB}(Ag_i) = (T, A)$, where in an initial state $T$ in each local instance $\mathbb{KB}(Ag_i)$, contains a copy of the upper Cyc ontology $\omega^U$ together with the perceptual anchoring ontology $\omega^P$, i.e. $(T) = \Omega : \{\omega^U \cup \omega^P\}$. During the operation of the agent, as soon as new information is anchored, the knowledge manager interacts primarily with the global knowledge base (Cyc), to retrieve the corresponding concepts and instances that describe grounded perceptual information by either creating new concepts or reusing previously defined concepts from past perceptual experiences.

The next step is to compile the logical representations of the agent’s perceptions and finally store them in a A-Box that is maintained in the Jena framework. Essentially this local reasoning component which can be seen as the knowledge management region shown closely in Fig. 6 (which contains the knowledge translation mechanisms and ontology fragment storage). To better explain what is actually done during the process and how data is represented let us consider the anchor example from § 2.1 Fig. 3. Its symbolic description contains the grounded predicates (AnchorID:UUID, Object:Book, colour:Red, Size:Big, Position:OnTable, SpatialRelation:leftOf_UUID2 ...). The knowledge manager queries about each predicate to find the corresponding concept and return the related available knowledge about it. For instance, when the system queries for the predicate ‘Book’, then the common-sense knowledge base returns the concept that represents the term (which is ‘BookCopy’) and its di-

---

\textsuperscript{8}(http://www.w3.org/)
\textsuperscript{b}(http://www.w3.org/RDF/)
\textsuperscript{i}(http://jena.sourceforge.net/)
\textsuperscript{j}(http://www.w3.org/TR/rdf-sparql-query/)
3.3.2. Global Semantic Anchoring Layer

The main challenge of integrating a large KR&R system like Cyc is to be able to synchronize the information with the perceptual data coming from multiple sensors originating from different agents, which is inherently subject to incompleteness, glitches, and errors. The anchoring framework provides stable symbolic information, despite fluctuating quantitative data of the perceptual grounding processes. Nonetheless, instances of objects must be created in Cyc and as objects can be dynamic (e.g., in the case of changing properties) proper updating of information needs to be managed. Ultimately, to enable the synchronization between the KR&R and Anchoring layers, three aspects are considered: defining anchoring within the context of the KR&R, handling of assertions, and handling of ambiguities. First we use the micro-theory concept (introduced in §3.2) to define the Perceptual Anchoring Ontology, so as to capture and combine all the concepts relevant for perception and anchoring, into one context. It is important to define the context of anchoring as a Micro-theory, as Cyc used this concept to differentiate
between contexts and also to maintain local consistency while eventually maintaining globally inconsistent assertions. Essentially with the PerceptualAnchoringMt, we model the perceptual domain that we wish to cover and hence the perceptual realm we wish our robot to operate on. Through the PerceptualAnchoringMt it is possible to express concepts about objects that are currently present in the anchoring space (and implicitly the environment), into the hierarchically structured knowledge of the KB and concurrently inherit all the common-sense information about this knowledge. For instance, if the location of ‘cup_1’ stored in an anchor, is the ‘Kitchen_1’, then the cup and the kitchen are instantiated as new concepts into the Micro-theory, inheriting all the properties as is-A related concepts or generalizations of the concepts Kitchen and Cup, such as ManMadeThing or HumanlyOccupiedSpatialObject etc. Furthermore when defining the PerceptualAnchoringMt, we introduced or reused the concepts, relations and descriptions of the different elements taking place during the whole perceptual anchoring process like: PerceptualAgent, IntelligentAgent, Kitchen, CupBoard, DishTowel, ...and relations like: is-A DishTowel SpatiallyDisjointObjectType, is-A DishTowel ExistingObjectType, genls DishTowel ClothTowel. We also introduced the concepts and sentences about the perceptual agents and their perceptual abilities: is-A Agent_2 IntelligentAgent, is-A Agent 2 PerceptualAgent, is-A Agent_2 LocalisedSpatialThing, hasAbility Agent_2 Vision, genls Perceiving Seeing. Furthermore to assist the PerceptualAnchoringMt, logical formulae were asserted. Examples of such formulae include (translated to English...
(1) PerceptualAnchoringMt is a general Cyc Micro-theory.
(2) Everything true in “things known”, independent of context, is also true in PerceptualAnchoringMt.
(3) Everything true in Cyc’s knowledge of general truths is also true in PerceptualAnchoringMt.
(4) If through experience an intelligent agent perceives a ‘SENTENCE’, then that ‘SENTENCE’ is a True Sentence.
(5) If through experience an intelligent agent perceives a ‘SENTENCE’ then that intelligent agent knows that ‘SENTENCE’.
(6) If some localized spatial thing is located in some other localized spatial thing ‘LOCAL’ and some other spatial thing ‘REGION’ contains ‘LOCAL’, then that localized spatial thing is located in ‘REGION’.
(7) ...

For instance points 2 & 3 are used in order to inherit all the concepts and facts that Cyc knows about the world. Point 4 (second-order Assertion) is used to make the agents’ perceptions true inside the PerceptualAnchoringMt. Finally the concept of an anchor and corresponding statements were defined (e.g. an anchor is a data structure, it indicates some instance of an object, which has some properties like colour, location, spatial relations, ..., which properties can permit some certain values HSV, units, ...).

3.3.3. Ambiguity resolution
Ambiguities may arise during the synchronization process and are resolved with the aid of the denotation tool via the natural language interaction with the user, as for example in the perceptual knowledge evaluation scenario presented in § 5.2.1. The denotation tool is an interactive step during the grounding process where the user is asked to disambiguate the symbol that is grounded between all the concepts in the knowledge base that denote this symbol. For instance, when grounding the symbol Flower and upon asserting the knowledge about this concept, an ambiguity will arise as the knowledge base contains multiple concepts denoting the term Flower; of which possible options can be: CutFlower, Flower-BotanicalPart, FlowerDevelopmentEvent, FloweryPlant. The user is asked to choose one of all those concepts, which are sorted by the probability of relevance or create a new one.

4. Instantiation of the Perceptual Anchoring Framework
We have implemented our approach in a distributed framework for easily interconnecting and enabling communication between different physical or computational
components. The framework is called Peis for Physically Embedded Intelligent Systems and it has been used in the context of ubiquitous robotics\cite{22,36}, distributed cognitive architectures\cite{14} and also used in prior research on cooperative anchoring\cite{23}.

The implemented anchoring architecture is built upon the Peis communication framework that allows the dynamic connection between the different components described in the previous sections. A component is any computerized system interacting with the environment through sensors and/or actuators and including some degree of “intelligence”. Each component is called a Peis and a Peis-Ecology is the collection of all those components. In the realization of the Peis-Ecology, the Peis relies on a distributed middle-ware that implements a distributed tuple-space on a P2P network: Peis exchange information by publishing tuples and subscribing to tuples, which are transparently distributed by the middle-ware. Each Peis also provides a set of standard tuples, e.g., to announce its physical appearance or the functionalities that it can provide.

As part of the test-bed, we use a physical facility, called the Peis-home, which looks like a typical bachelor apartment of about 25 m$^2$. It consists of a living room, a bedroom and a small kitchen as shown in Fig. 8 (left). The Peis-Home is equipped with communication and computation infrastructure along with a number of sensors like camera and localization components. In our ecology there is an interactive robotic system based on an ActiveMedia PeopleBot platform for indoor research. In addition to the usual sensors, the robot is equipped with a SICK LMS200 laser range finder and with a Sony PTZ colour camera. The integration in our current approach involved the mobile robot which is capable of perceiving the world through different typical heterogeneous sensors and is acting as the Mobile Anchoring Agent. Also an immobile robot (or better, Ambient Anchoring Agent) which manifests through sensors and actuators embedded in the environment like observation cameras, localization systems, RFID tag readers, mechanical manipulators etc. Finally, a desktop computer is used to undertake all the computational and memory demanding tasks related to the common-sense knowledge base (Cyc).
4.1. Perception

In the introduction (§ 1), we have stated that the context of anchoring is to act as the mediating link between perception and knowledge representation. In the perceptual layer we describe the implementation of the two kinds of perceptual processes, Vision and Localization. The perceptual systems gather information from the sensors which is then filtered by appropriate feature extraction and matching mechanisms. Perceptual Information is grouped according to the sensors (camera, range-finder, sonar, etc.) that are mounted on each physical agent. In our case we are using the two perceptual agents described above, the mobile robot and the ambient anchoring agent. Information from each perceptual agent, is then fed into the anchoring processes. The interchange of data is achieved through the Peis-middle-ware where each module is publishing information to or reading information from the tuple-space.

4.1.1. Vision

The visual perception that is used on all the agents consists of pairs of Peis-components, one camera component that provides images to the Object Recognition component. In our preliminary approach we had adopted a SIFT (Scale Invariant Feature Transform)\textsuperscript{28} based computer vision algorithm for recognizing typical objects in the environment. Briefly, in the previous object recognition algorithm there was lack of support for training of new objects on-line\textsuperscript{k} while the implementation of SIFT we had adopted was fast enough to enable recognition. However the SIFT features were found to be robust against illumination, scale and rotation invariance while still achieving a robust and stable recognition of objects over time irrespective of the number of objects that were added to the trained SIFT database. Hence, we continue exploiting SIFT features by developing an updated faster and more flexible algorithm from the one which can be found in previous work\textsuperscript{11}.

The steps of the algorithm are shown in Fig. 9. Initially the camera pushes streams of images to the Object Recognition, which extracts the SIFT features and matches them against the database of features which contains the trained objects. If there is a request from the user to train an new object, we segment instead the region that the user specified to train and we associate the newly defined SIFT key-points with either an already trained object (if it is seen from another viewpoint for example) or create a new object in case the object is trained for the first time. Then we prune the matches for each object, by distance, where we start reducing the outliers by enforcing a best-to-second-best key-point match ratio. Finally as it is suggested by Lowe, we detect and prune the outlying matched features by using Hough Transform voting for cluster identification to search for keys that agree upon

\textsuperscript{k}on-line training indicates the ability to train a new object without having to suspend the vision of the robot in order to train an object. Therefore training is done simultaneously with recognition, in the improved version of the object recognition component
Visual Percept
Capture Frame
Extract SIFT
Match
Train
Segment Recognized Regions
Prune by Distance
Hough Transform
... 

Fig. 9. Diagram of the different steps of the vision algorithm. SIFT\textsuperscript{28} features are extracted from the image data coming from the camera sensor. The features are matched against the features database and further filtered so as to find the best object model and generate the visual perceptual signature of the object.

a particular model pose and Affine Transformation so as to relate the model to the image with a linear least squares solution\textsuperscript{28}. Finally, we capture the robustly segmented regions in which there are objects recognized. The output of the algorithm is the perceptual signatures, one for each object, which contain the segmented extracted image, one colour normalized copy of it, estimated depth of the object, its unique identifier and information about the scale, orientation, reference position in the camera frame, error, probability of match and other perceptual information.

The SIFT implementation we use as the core processing engine is the Sift-GPU\textsuperscript{146}. It is a general-purpose computing, on graphics processing units, implementation of the SIFT algorithm. It processes pixels in parallel to build gaussian pyramids and detect DoG key-points. Based on GPU list generation SiftGPU uses a GPU/CPU mixed method to efficiently build compact key-point lists. Matching is done through the GPU exhaustive/guided sift matcher included with the library. It multiplies the descriptor matrix on GPU and finds the closest feature matches on GPU. Between the three (GLSL/CUDA/CG) implementations we used the CUDA implementation. We achieved a stable recognition at appx. 17 frames per second in 640 × 480 pixel resolution and approx. 13 frames per second in 800 × 600 pixel resolution, with two standard NVIDIA Geforce9800GT, while maintaining 20 trained objects (from multiple perspectives) in the database. In cases where 3 or 4 objects are simultaneously recognized the performance degradation was not noticeable. Therefore for the evaluation we used the 800 × 600 pixel resolution throughout the test runs as greater resolution implied more detail and in turn eventually more SIFT key-points, increasing the recognition of the object while being able to operate

\textsuperscript{1}http://www.cs.unc.edu/~ccwu/siftgpu/
still in relative real-time (i.e. approx. 15 frames per second).

4.1.2. Localisation

In our approach, vision and localization are fused together to contribute on grounding extra semantic information to the anchors, more specifically regarding the spatial distribution of the objects. So we have developed a localization algorithm which is shown in Fig. 10, and is using self localization with coordinate system transformations in accordance with the approximated depth data from vision to (a) represent the agent itself and (b) the objects the agent perceives, in a global 3-D Coordinate System. It differs only according to the nature of the agent. For instance, in the mobile robot the self localization component is based on the standard Adaptive Monte-Carlo Localization algorithm, while for the ambient anchoring agent the self localization is fixed to the point where each camera is mounted in the environment. After each agent performs self localization, position and pose information is pushed into the grounding process, in order to ground the concepts and then sentences of the location and pose of each agent. For example, from the global coordinates \((x:1 \text{ m}, y:1 \text{ m}, z:1.5 \text{ m}, \text{angle:65 deg.})\) which are the output of the mobile robot’s self localization algorithm, the symbol ‘LivingRoom_1’ is grounded for the predicate Location, producing the logical formulae: \(\text{ObjectFoundInLocation Agent_1 LivingRoom 1, ObjectIsFacing Agent_1 LivingRoomTable 1, …} \). Then, for each recognized object, information from the vision module is fused into the localization algorithm. Ac-

Fig. 10. Diagram illustrating the different steps in the coordinate system transformation and percepts fusion process. After the self localization of each agent, the visual percepts are fused with the topological features, so as to represent all the visual objects in one global coordinate system.
According to the approximated depth, we compute the translation of the object from the coordinate system of the camera, initially to the local coordinate system of the agent (e.g. with reference to the robot) and then to the global coordinate system with reference the global environment map. These transformations are needed in order to represent ultimately everything in one unified global coordinate system, and provide compatible and robust data for computing the qualitative spatial relations, while also aiding during the unification phase in the global anchoring process.

4.2. Grounding Relations

With respect to the grounding relations, in our current approach we have implemented mostly spatial and appearance related algorithms, to try to capture an adequate symbolic description for the objects in the environment. Briefly we implemented the following grounding relations: colour (i.e. ‘red’), topological localization ('in the kitchen'), spatial relations ('to the left of'), object category ('cup'), visibility ('visible'). Some of them are already discussed, such as the object category which is based on the vision algorithm, or the topological localization during the coordinate system transformation processes. The grounding of colour, visibility and spatial relations have already been discussed in our previous work\textsuperscript{11,27}, therefore the details are omitted, except for the spatial relations which we further develop in the present work. A new model for representing and computing spatial relations is based on qualitative spatial reasoning principles and allows us to explore both egocentric and allocentric spatial relations of agents and objects in the environment.

4.2.1. Spatial Relations

Typically, in natural language interfaces for service robots the goal is usually specified with the class name of the object. In more open scenarios especially in the common-sense domain, the user may eventually need to interact with the robot using spatial expressions in order to disambiguate, refer to, learn or reason upon objects (i.e. the book on the table, the blue box on the left of the pasta is a box with cereals). One of the main motivations behind qualitative spatial representation is the way humans conceptualize spatial information and this information is conveyed through natural language. An obvious advantage of qualitative spatial representation is the handling of imprecise information. Using qualitative relationships it is possible to express only as much information is necessary or perceived. The level of precision that can be represented depends on the granularity of the qualitative relations. Therefore we focus on a strategy for computing the spatial relations, as we already have previously adopted and investigated a model for inferring qualitative spatial relations within the context of perceptual anchoring\textsuperscript{11,32}. We have further developed the algorithm by combining ideas from Star Calculus in Qualitative Spatial Representation\textsuperscript{35} and the Extended Panorama Representations from Qualitative Spatial and Knowledge Representation\textsuperscript{45}. Star calculus is a class of direction calculi which are a generalization of several existing calculi for representing
and reasoning about the qualitative direction between points in a plane, offering arbitrary level of granularity\textsuperscript{35}. Spatial knowledge based on sensory input has to be represented either in terms of 	extit{egocentric} or 	extit{allocentric} spatial relations. Allocentric and egocentric spatial relations refer to the reference system, which they rely on. Since perception is egocentric by nature, any allocentric spatial relation has to be deduced indirectly from egocentric relations. In an egocentric representation, spatial relations are usually directly related to an agent by the use of an egocentric frame of reference in terms as left, right, in front of, behind. On the other hand, representations based on an allocentric frame of reference remain stable but are much harder to acquire\textsuperscript{35}. Additionally, the number of spatial relations, which have to be taken into account, may be much larger because we have to consider the relations between each object and all other objects in the environment, whereas the number of relations in egocentric representations can be significantly smaller. In addition, egocentric representations provide better support for rotation and translation invariant representations when used with a qualitative abstraction.

A clear disadvantage when an agent uses or communicates an egocentric spatial expression is that the expression cannot be used by another agent or a human without a transformation or at least the use of an intermediate allocentric frame, which gives a common frame of reference. The notion “\textit{frame of reference}” can be thought of as labelling distinct kinds of coordinate systems. Linguistic literature usually invokes three frames of reference: an intrinsic or object-centred frame, a relative (deictic) or observer-centred frame, and an absolute frame.

In our use of spatial relations, we will assume a relative reference frame that is based on the robot’s point of view. We are also capturing relations between objects with an intrinsic reference frame, i.e., an inherent front or rear that is defined by the object itself. Objects are represented by an idealized point location, derived by
Local Spatial Relations

- **A14r**: A is near and in front of robot r.
- **B12r**: B is near and in front but somewhat to the left or robot r.
- **C6r**: C is near and behind robot r.
- **r14**: Robot r is near and behind A.
- **r12**: Robot r is near and behind but somewhat to the left of B.
- **r6**: Robot r is near and in front of C.

Global Spatial Relations

- **A14Agent1**: A is far and in front of Agent1.
- **A14Agent2**: A is far and in front of Agent2.
- **B12Agent1**: B is far and in front of but somewhat to the left of Agent1.
- **B12Agent2**: B is far and in front of but somewhat to the left of Agent2.
- **C14Agent1**: C is far and in front of Agent1.
- **C14Agent2**: C is far and in front of but somewhat to the right of Agent2.
- **D8Agent2**: D is far and behind but somewhat to the left of Agent2.
- **D12Agent1**: D is far and in front of but somewhat to the right of Agent1.
- **A2B**: A is far and to the right of B.
- **A14C**: A is far and in front of C.
- **A0D**: A is far and in front of but somewhat to the right of D.

Projecting the agent’s point of view onto the plane of the object’s centre of gravity. At the local level, perceptual data (geometrical representations from the camera-to-local coordinate system transformation § 4.1.2) from the localization module describing allocentric spatial distribution of recognized objects (visible and not visible) are represented as an Extended Panorama Representation as seen in Fig. 11 (a), for every agent. So that we can use the Star Calculus \( \text{STAR}_{[30, 60, 120, 150]}(30) \) (Fig. 11 (b)) using the angles between the centre of view of the robot and the objects that are distributed around the agent, and ground the symbolic binary spatial relations by combining a three-part linguistic spatial description: (1) a primary direction (e.g., the object is in front), (2) a secondary direction which acts as a linguistic hedge (e.g., but somewhat to the right), and a third part which describes the Euclidean distance between the object and robot (e.g., the object is near). Two classes of binary spatial relations between a reference object and the object to be located (located object) are considered: the topological relations **at**, **near** and **far**, and the projective relations **in front of**, **behind**, **right**, **left**, **above** and **below** and discretion mixes of those using the secondary direction, while applied to the 3D space in the panorama surrounding the robot. Examples of the generated intrinsic and extrinsic relations on the local level are shown in Fig. 11 (a), which at the local level, are sufficient, to let us express things like “Object C is located near and behind of me (the mobile robot)” in the local knowledge base, by grounding and attaching the spatial relations “back” as primary direction and “near” as the proximity relation, to the anchor representing the object C. Finally at the global anchoring space where all anchors/objects are unified, the geometrical information of all the anchors is translated and fused into a global coordinate system (which coincides...
with the map of the environment), through the coordinate system transformations (local-to-global) mentioned in § 4.1.2. Since information is maintained from the local anchoring space, we inherit all local spatial relations of each agent we described above. In addition we extend the global symbolic descriptions by grounding also the intrinsic (object-centred reference system) and extrinsic spatial relations using the fused global anchor information about anchors that are perceived from different anchoring agents. In this way objects are spatially referred to other objects (e.g. “Object X is near and to the left of Object Y”, and contrary, “Object Y is near and to the right of Object X”) so that we can expand the domain of spatial relations to include both egocentric and allocentric spatial expressions, ready to be asserted and reasoned upon, in the knowledge base. For practical reasons the egocentric point of reference is the one of the mobile robots in a way similar to how a person would communicate with another person using spatial relations with reference the other person, instead of any other point of reference. With adding allocentric spatial relations, we can now globally represent in the common-sense knowledge base, statements of the form “Agent A Sees that Object X is near, in-front and above Agent B” where Agent A is the ambient anchoring agent in the living room, Object X is the painting on the wall as it is seen from the ambient agent’s camera in the ceiling, and Agent B is the mobile robot which does not see the painting at all. In a such scenario we have achieved to reason upon egocentric perceptual information that was perceived by another anchoring agent’s perceptual system. So, when the mobile robot perceived a book in front of it, and by having the knowledge that there was a painting above from the perception of the ambient agent, it could successfully infer that “the book is below the painting” as well as “the painting is above the book” for the anchors concerning the Painting and the Book, originating from different anchoring agents.

4.3. User Interface & Natural Language Interaction

In a complex multi-modal system as the one we are describing in this work, there are different kinds of information we wish to monitor so as to be able to evaluate the system. For this reason we have developed a multi-modal monitoring and interaction interface (Fig. 13) which has three ways for accessing information throughout the system’s operation. It provides a primitive natural language interface to interact with the different agents and the knowledge base (Fig. 13(a)), displays the state of the whole framework (Fig. 13(b)) and provides runtime information about the performance of anchoring (Fig. 13(c)). The main component of the tool, the 3D visualization environment, is the main observation feature for monitoring anchoring. It is a scene rendered in a 3D engine, of the environment that encapsulates every non perceptual object in the environment such as the wall surroundings, the large and small objects like cupboards, and the agents themselves. Then information is extracted from the global anchoring space to reconstruct virtually the perception of the agents by adding and updating the anchored objects in the rendered scene. The
user can navigate freely in the virtual environment to observe the positions of the different agents and anchors in the environment. The 3D model was implemented as a scene-graph using the Java3D\textsuperscript{m} and Java Open Dynamics Engine\textsuperscript{n} libraries, and is attached to an interface where it extracts perceptual and symbolic information from the global anchoring space. Besides navigating and monitoring, the user can also interact with the mouse to click on different objects on the virtual environment and trigger the other part of the monitoring tool which is the runtime monitoring panel as seen in Fig. 13 (c). As soon as the user selects to monitor an anchor, the runtime panel displays all the information about this anchor, as it varies over time. This includes both perceptual information (such as the image of the object, reference coordinates, translated global coordinates, relative scale and orientation translation, colour information) and symbolic information (such as the unique identifier, the object category, the colour, location). Runtime information about the different agents (such as their localization information, spatial relations) is displayed as well.

The final part of the interaction tool, is the command line and natural language interface (Fig. 13 (a)). We have implemented a simplistic template based Natural Language parser which is able to handle two categories of natural language sentences. The first is the different commands with which the user has access to

\textsuperscript{m}https://java3d.dev.java.net/
\textsuperscript{n}https://odejava.dev.java.net/
Table 1. Natural language template parsing examples

<table>
<thead>
<tr>
<th>NL Query</th>
<th>Parsed Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>-What do you see?</td>
<td>query:(sees ?AGENT ?OBJECT)</td>
</tr>
<tr>
<td>-Where is the red cup?</td>
<td>query:(objectFoundInLocation (and (isa ?CUP Cup) (objectHascolour ?CUP RedColour) ?WHERE))</td>
</tr>
<tr>
<td>-Is the red cup in the kitchen?</td>
<td>query:(ObjectFoundInLocation (and (isa ?CUP cup) (objectHascolour ?CUP RedColour) Kitchen_1))</td>
</tr>
<tr>
<td>-Is the red cup used for drinking?</td>
<td>query:(ObjectUsedFor Drinking (and (isa ?CUP cup) (objectHascolour ?CUP RedColour)))</td>
</tr>
<tr>
<td>-What colour is the flower?</td>
<td>query:(ObjectHascolour (isa ?FLOWER Flower) ?WHAT)</td>
</tr>
<tr>
<td>-What do you see to the left of you?</td>
<td>query:(Sees (isa ?AGENT PerceptualAgent) (leftOf ?WHAT ?AGENT))</td>
</tr>
<tr>
<td>-Is the coffee on the right of the washing liquid?</td>
<td>query:(rightOf (isa ?COFFEE Coffee-Instant) (isa ?WASHINGLIQUID WashingLiquid))</td>
</tr>
<tr>
<td>-Do you know that the green flower is located in the kitchen?</td>
<td>query:(knows (isa ?AGENT PerceptualAgent) (and (isa ?FLOWER Flower) (objectHascolour ?FLOWER GreenColour) (objectFoundInLocation ?FLOWER Kitchen)))</td>
</tr>
</tbody>
</table>

the contents of the knowledge base. Some of the commands which are present are: Translate, for translating (probing) a sentence from English to logical representation and vice versa. Guess, is used to search for a Cyc concept that denotes the word we are trying to guess (i.e. guess CD returns, CompactDiskGeneric, ...). Ask, first translates the sentence from English to logical representation, and queries the knowledge base with it allowing queries with variables or without. Statistics, shows anchoring statistics, on the command line interface, in text form. Name, returns a random specialization of a concept (i.e. name an Agent, gives HumanAgent, or IntelligentAgent for instance). Make, asserts that the concept is another concept (i.e. make the BlueBox, a Book), Kill, deletes a concept. Explain is used for displaying the proof graph for the last query. Create Individual and Create Collection are used to create new concepts either instances or classes in the knowledge base, while Assert and Un-assert a sentence, similarly translates and asserts or retracts the sentence from the knowledge base. Finally Comment is used to provide comments for the concepts created in the knowledge base. To translate from logical representation to English, we use the built in natural language generation and linguistic features of Cyc. The other class of sentences the parser is able to capture is allowing the user to type in questions or complementary assertions in natural language using predefined templates. All the occurrences of the the word “you” are reformulated to the agent’s name while the questions parser, is able to capture 15 categories of “what” questions, 19 categories of “who” questions, 7 categories of “is ... ” questions and 18 other categories, with and without variables. Some examples of the Natural Language Parses for both commands and NL questions are better shown on Table 1. While from a technical point of view, communication occurs between the human user and the Peis-Ecology, it is presented to the user as an interaction between the user and the robot, as this interaction has shown to facilitate a better HRI9.
5. Experimental Scenarios

To evaluate the implemented architecture we follow a two fold strategy. The first section examines the performance of the whole anchoring system and the statistics extracted during execution, while the next section focuses on the knowledge representation aspect, exploiting different scenarios that illustrate the benefit of using a common-sense knowledge representation system. Both parts refer to one single run which takes place in the PeisHome, utilizing the Peis Framework to allow communication between the physical components and the Anchoring Framework. We have trained the vision algorithm with 16 objects (samples shown in Fig. 14), via the on-line training tool described in § 4.1.1.

Training was done by placing the different objects in front of the robot while selecting them through the monitoring interface using the mouse to specify the region where the object is located. The total training time for all the objects was 10.88 seconds, while on average the algorithm needed 680 ms to train each object from 4 view points. We then placed the objects around the PeisHome in a random manner, covering a great amount of different combinations of spatial relations. We allowed the robot to navigate\(^{a}\) around the environment so as to recognize and anchor the trained objects, while in the meantime the ambient anchoring agent anchored its own set of objects. We allowed the user to communicate with the system by typing into the console based graphical user interface, questions (related to robot’s perceptions, or common-sense), or communication commands in natural language according to the process described in § 4.3 so as to manipulate the contents of the knowledge base. We use the multi-modal interaction interface to extract the statistics and to be able to observe and compare that what has been virtually reconstructed after the anchoring processes, reflects the current state of the environment. Through the examined case studies we illustrate the ability of the KR&R component to accept queries and interaction commands and return interpretable information.

\(^{a}\)through predefined way points, with obstacle avoidance ‘on’
5.1. Perceptual Anchoring Framework Performance

In this part of the evaluation of the implemented system we are mostly interested to extract information related to how well the system performed, when recognizing, anchoring and maintaining in time the anchors referring to the objects in the environment. In Fig. 15 and Fig. 16 we see a map showing how the objects are anchored and more specifically their position information from the unified global anchoring space. Agent_1 which is the Ambient Perceptual Agent, remained static as its visual sensor remained fixed in the ceiling. It also includes the trajectory for the path that Agent_2 (which is the mobile robot) followed, starting from the living room area until the kitchen area. The clusters of objects indicate the approximated global object coordinates. The error is because of motion blur and is generated during the movement of the mobile robot, when approximating the depth of the objects and implicitly the geometrical transformations that represent the objects in the global reference frame. However we see from the density of the points that despite the errors, once the mobile robot’s vision stabilized, the approximations become more stable as well.

We also see some indication of the performance of the vision systems that run on each perceptual agent. In Fig. 17 we see the evolution of different categories of anchors over time. Initially there are the individual local visual percepts from Agent 1 which were stably recognized over time, according to the 4 trained objects in the agent’s vision system (Chair, Sofa, TV, and Painting). The visual percept of Agent 2 initially are zero as the agent was not moving in the beginning of the experiment.
Fig. 16. Scatter plot showing how the anchors and the agents are distributed in the metric map of the environment, from a 3D perspective (rotation=−24°, elevation=16°).

Fig. 17. Plot showing different anchoring runtime statistics over the duration of the experiment. It includes the total number of anchors acquired, the active anchors (used for the global spatial relations) as well as both agents’ visual percepts number.
As the agent starts exploring the environment while moving from the living-room to gradually reach the kitchen, its percepts vary according to the number of objects it perceived, with a maximum of 4 objects simultaneously recognized. In the same plot we also see the total number of anchors created and maintained in the global anchoring space, where the number gradually increases up to the 16 objects in total, which we initially trained. Finally the number of active anchors indicates the number of anchors that are participating in the computation of global spatial relations. Here we isolate essentially the anchors that correspond to objects that are located in the same topological region that the mobile robot (reference point) is located, whether they have been originally perceived by the robot or the ambient agent. For instance, when the mobile robot is operating in the living room, the number of active anchors is the number of anchors from the global anchoring space that correspond to the objects perceived by the mobile robot and the ambient agent together. We are not interested in computing the spatial relations about objects located in the kitchen or bedroom, when the robot is located in the living-room, because the reference point is not located in those topological regions. It is interesting to note that the two perceptual systems provided complementary perceptual information that was cooperatively acquired from different sensors while maintained over time, and also that this information is stable enough to be streamed into the KB as we see in the next section. Except from the approximated positions of the anchors we see also samples taken from the 3D interface (Fig. 18) which were used to allow us to navigate in the virtual reconstructed environment and evaluate how the anchors were created and evolved throughout the experiment in the virtual environment. By using the monitoring interface (Fig. 13) on each anchored object we evaluated that the grounded properties correspond to the real perceptual properties of the objects, as when grounding the colour, visibility or spatial relations.

5.2. Knowledge Representation and Reasoning Evaluation

The advantage of using a common-sense knowledge base is that we can exploit the information encapsulated so as to not only reason about things that were directly asserted by grounding percepts to symbols, but also: (a) express and differentiate clearly between perceptual and epistemic knowledge, (b) communicate (using) this knowledge or (c) expand it. Hence our focus is on the communication both between the perceptual agents and between the agents and the user. During the anchoring scenario described in § 5.1, besides the performance of the system, we simultaneously investigated possible cases exploiting the knowledge representation system, through natural language and commands, using the interaction interface and the different permitted commands as described in § 4.3.

5.2.1. Perceptual Knowledge

Perhaps the most obvious scenario is the one where, the knowledge from the perceptual anchoring agents populating the Cyc knowledge base within the Percep-
Fig. 18. (a) Sample of different components of the system (localization component of the mobile robot (left), output of the vision component with extracted SIFT features (middle), Observation view of the robot perceiving objects in the virtual environment) from the experiment execution where the robot is located in the living room looking at the two books on the living room bench. (b) Sample where the robot is located in the kitchen observing a box of cereals and a box of pasta and (c) Overview of the virtual environment where there are all the anchored objects and the two agents at the end of the execution.

Cooperative Knowledge Based Perceptual Anchoring

$t_{\text{context}}$, can be used to enable inference upon the instantiated perceptual concepts and sentences. It is possible to use the object’s category, colour, spatial or topological relations, along with other grounded perceptual information, to (1) refer to objects, (2) query about objects and (3) infer new knowledge about objects. Through simple natural language questions, as permitted by the template based parser, we examined possible interaction scenarios, examples of which we present on Table 2. We see the user’s questions, the parsed output which contains the concept bindings, and the system’s response. Initially we see queries regarding perceptual information and properties. Due to the expressivity of Cyc we were able to differentiate perceptual and epistemic knowledge, but at the same time keeping both coherent in the KB. As we see in the example, the system responds to both queries of “Do you know that … ?”, and “Do you perceive (or see) that … ?”. This virtually creates the effect of memory, where the agent knows about things seen in the past (epistemic knowledge), but currently not perceived (perceptual knowledge). For what is not perceived any more, the perceptual information is not updated but rather maintained, based on the latest grounded estimate. We can use both (egocentric & allocentric) spatial relations to refer about objects, while at the same time we are able to reason about them as well, through the common-sense rules in the knowledge base (as we see in parts of the dialogue).

Another aspect we explored concerns queries about the perceptions of each agent independently. We were able to combine queries with the anchoring find functionality to locate the “Coffee” or the “Blue Book”, by letting the robot to execute the query and use find to Re-acquire the anchor that satisfies the query, by converging into an anchor from the anchoring space that corresponds to the specific object.
Finally in the same context, we were also able to use common-sense information already inherited from the KB, so as to infer if a concept instance satisfies the initial query (e.g. when asking if objects are used for eating).

5.2.2. Switching context

In this scenario we show the use of common-sense knowledge in multiple contexts (i.e. in our case MicroTheories), through the multi-context inference capabilities of the Cyc system. By default all the NL queries are evaluated within the PerceptualAnchoringMt, except if the context is specified explicitly through a linguistic pattern. The examples on Table 3 show questions entered by the user which are evaluated for any possible context where this query can be satisfied. In our case there are two contexts where the query (ObjectFoundInLocation BookCopy ?Where) is satisfied. First it is true for Libraries through the assertion (objectFoundInLocation

<table>
<thead>
<tr>
<th>User</th>
<th>How many Objects do you See?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self</td>
<td>I See 6 Objects. 4 Objects from Agent1. 2 Objects from Agent2.</td>
</tr>
<tr>
<td>User</td>
<td>Do you perceive, a Book?</td>
</tr>
<tr>
<td>Self</td>
<td>I See, 6 Books.</td>
</tr>
<tr>
<td>User</td>
<td>What colour is the Book?</td>
</tr>
<tr>
<td>Self</td>
<td>I See, 6 Books, Which Book?</td>
</tr>
</tbody>
</table>

User   > How many Objects do you See?  
Self   > I See 6 Objects. 4 Objects from Agent1. 2 Objects from Agent2.  
User   > Do you perceive, a Book?  
Self   > I See, 6 Books.  
User   > What colour is the Book?  
Self   > I See, 6 Books, Which Book?  

User   > What colour is the Book, located to the right of Coffee?  
Self   > 1 Book Located to the right of 1 Coffee. The Book has colour Blue.  
User   > Is the Food Located in the Kitchen?  
Self   > True. 2 Food Located in 1 Kitchen. Cereals is Located in the Kitchen. Pasta is Located in the Kitchen.  
User   > Do you Know that, the Book is Located in the Living Room?  
Self   > I Know, 2 Books Located in the Living Room.  
User   > Do you perceive that, the Book is Located in the Living Room?  
Self   > False.  
User   > Ask Agent1: Is the Cereals to the left of Pasta?  
Self   > True.  
User   > Ask Agent2: Is the Cereals to the left of Pasta?  
Self   > False.  
User   > Find the Coffee in the Kitchen.  
Self   > Found 1 Coffee-Instant Located in the Kitchen. Searching ...  
User   > Find a Book.  
Self   > Found 6 Books. Which Book?  
User   > Find the Blue Book in the Kitchen.  
Self   > Found 1 BookCopy has colour Blue and Located in the Kitchen. Searching ...  
User   > What is a Book?  
Self   > A collection of InformationBearingObjects. A book is a physical obj. ...  
... Physical information bearing object, Kind of artifact not distinguished by brand or model, Spatially disjoint object type.  
User   > Is the Book used for Eating?  
Self   > False.  
User   > Is the Pasta used for Eating?  
Self   > True.
Table 3. NL interaction regarding different contexts.

<table>
<thead>
<tr>
<th>User</th>
<th>What is a book, in any context?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self</td>
<td>1. A Book is located in the LibrarySpace, in the Buildings context.</td>
</tr>
<tr>
<td>Self</td>
<td>2. A Book is located in the LivingRoom, in the PerceptualAnchoring context.</td>
</tr>
<tr>
<td>User</td>
<td>What colour is the coffee, in any context?</td>
</tr>
<tr>
<td>Self</td>
<td>1. Coffee has colour BrownColour, in the HumanPerception context.</td>
</tr>
<tr>
<td>Self</td>
<td>2. Coffee has colour BlackColour, in the PerceptualAnchoring context.</td>
</tr>
</tbody>
</table>

BookCopy LibrarySpace) in the context containing knowledge about Buildings (Buildings Mt) and the Living-room through the assertion (PerceivesThat Agent_1 (ObjectFoundInLocation BookCopy_1 LivingRoom_1)) in the Perceptual Anchoring context (PerceptualAnchoring Mt). In the same spirit as we see in the following query, asking about the colour of coffee, with the linguistic template “in any context?” results in the parsed query, (and (ObjectHascolour Coffee-Ground ?What) (isA ?What Colour), which is satisfied again in two contexts. In the Human Perception Mt context, according to the knowledge (typicalcolourOfType Coffee-Instant BrownColour) and the generalisation (genls Coffee-Instant Coffee-Ground). While in the Perceptual Anchoring Mt context, the query is satisfied when the binding ‘?What’ evaluates to ‘Coffee_1’ and with the following assertions: (perceivesThat Agent_1 (ObjectHascolour Coffee_1 BlackColour)) and (isA Coffee_1 Coffee-Ground). Therefore according to the answers, coffee is typically brown as perceived by humans, however in the robot’s perceptual anchoring context, the only instance which actually refers to the box of coffee in the kitchen, is perceived to be black (as it is the colour of the box that contains the coffee). Hence through this scenario we exploit reasoning with (a) different contexts, (b) contradicting statements (but in different contexts, see § 3.2) and (c) already available common-sense knowledge (Table 3).

5.2.3. Knowledge exchange

In this case we show how the different perceptual agents can communicate using logical sentences, in order to exchange information between each other in a cooperative manner. In this particular scenario we focus on the local knowledge base of the perceptual agents. Thus we trigger the behaviour in every agent to query other available agents so as to complement its perceptual knowledge and answer queries. The communication format is in SPARQL queries and KB operations. The goal is set by the user when asking a specific agent (Agent 2) about a specific object. This agent in question fails to perceive this object, so the query is relayed to all other available perceptual agents. In this case the ambient perceptual agent (Agent 1), which in turn evaluates the query and delivers the result back to the requester (Agent 2) who answers the question.

As seen in the dialogue on Table 4, the first agent (Agent_2) communicates with the ambient agent (Agent_1) so as to answer the query. Such ex-
change of information may not only be used for satisfying NL queries but also in other cases such as if a plan had to be executed. Consider a scenario in which every morning the planner of the smart home wants to ensure that the Cereals need to be in the Kitchen for the preparation of the breakfast, so as in case they are not, the planner might want to add it to the shopping list. So at some point in the plan processing the following query arrives at the ambient anchoring agent: (and (isa ?Cereals BreakfastCereal) (ObjectFoundInLocation ?Cereals Kitchen_1)). But for some reason (i.e. the box of cereals was fully occluded from the view of the ceiling camera in the kitchen, of the ambient agent) the query failed, according to the local knowledge base as there is not an anchor regarding an object which is of type BreakfastCereal found to be in the Kitchen. So the ambient agent forwards this query to every other available anchoring agent (i.e. the mobile robot). The mobile robot evaluated the query in its own local knowledge base, resulting in the anchor of the cereals that has been observed the kitchen in the past. Therefore through the following assertions: (isa Cereals_1 BreakfastCereal) and (ObjectFoundInLocation Cereals_1 Kitchen_1), the query evaluated to true for the bindings shown below and the robot performed a find to re-acquire the anchor of the cereals. After the evaluation and re-identification
of the box of cereals, the updated perceptual knowledge was asserted and the query evaluated to true also for the ambient agent which received the result, and continued executing its planning process.

5.2.4. Acquiring new knowledge

In this scenario we discuss how it is possible to use the proposed system so as to acquire new knowledge or compensate for incomplete knowledge. This can be done via two ways, using the Natural Language Interface, or through the semantic web. The motivation behind this scenario lies on current approaches to align Wikipedia articles with Cyc concepts\textsuperscript{31} or to generate plans for robotic environments by gathering task descriptions from the semantic web using the WordNet database and the Cyc ontology\textsuperscript{42}. First using the different implemented commands (§ 4.3), the user is able to manipulate the concepts in the knowledge base as well as assert new knowledge, as for example on Table 6. First, we simulate incomplete knowledge in the knowledge base by removing the corresponding concept and related assertions. The system notices that the knowledge about a recognized object (pasta) is missing and uses the links of Cyc to other on-line ontologies and knowledge bases in the Semantic Web (i.e. DBpedia, UMBEL), in order to compensate for the lack of knowledge. The search returned two results with the owl properties $\text{owl:sameAs}$, one for the UMBEL ontology (http://umbel.org/umbel/sc/Pasta) and one for DBPedia (http://dbpedia.org/resource/Pasta). Then the related concepts with their corresponding descriptions are downloaded, parsed and asserted in the knowledge base. The user asserts that the initial blue colored object in the kitchen, is an instance of the learned concept Pasta using the make command, while then the user verifies that it was successfully integrated in the common-sense knowledge base, by
Table 6. Acquiring new knowledge from the semantic web.

<table>
<thead>
<tr>
<th>User</th>
<th>remove collection Pasta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self</td>
<td>I see an unknown blue object (36-UNKN-BLUE) in the Kitchen.</td>
</tr>
<tr>
<td>Self</td>
<td>found 1 DBPedia link: (Pasta owl:sameAs cyc:en/Pasta)</td>
</tr>
<tr>
<td></td>
<td><a href="http://dbpedia.org/resource/Pasta">http://dbpedia.org/resource/Pasta</a> retrieving available knowledge ...</td>
</tr>
<tr>
<td>Self</td>
<td>found 1 UMBEL link: (Pasta owl:sameAs cyc:en/Pasta)</td>
</tr>
<tr>
<td></td>
<td><a href="http://umbel.org/umbel/sc/Pasta">http://umbel.org/umbel/sc/Pasta</a> retrieving available knowledge ...</td>
</tr>
<tr>
<td>Self</td>
<td>create collection Pasta</td>
</tr>
<tr>
<td>Self</td>
<td>assert (isa Pasta (ArtifactTypeByGenericCategory, SpatiallyDisjointObjectType, ExistingObjectType ... ))</td>
</tr>
<tr>
<td>Self</td>
<td>assert (genls Pasta Food)</td>
</tr>
<tr>
<td>Self</td>
<td>assert (denotation Pasta-TheWord MassNoun 1 &quot;Pasta&quot;)</td>
</tr>
<tr>
<td>Self</td>
<td>assert (... ...)</td>
</tr>
<tr>
<td>User</td>
<td>make &quot;36-UNKN-BLUE&quot; Pasta in PerceptualAnchoringMt.</td>
</tr>
<tr>
<td>Self</td>
<td>assert (isa Pasta_1 Pasta Mt : PerceptualAnchoringMt.)</td>
</tr>
<tr>
<td>User</td>
<td>Is the Pasta in the kitchen used for Eating?</td>
</tr>
<tr>
<td>Self</td>
<td>True. 1 Pasta Located in 1 Kitchen. Justification Mt : BaseKB (objectUsedFor EatingEvent Food) &amp; (genls Pasta Food)</td>
</tr>
</tbody>
</table>

asking if that concept instance Pasta_1 can be used for eating. This returns True as “typically food is used for eating” and “Food is a generalization of Pasta” in the knowledge base. Thus the box of pasta in the kitchen Pasta_1 can be used for eating.

6. Related Work

In the literature related to the present work, we see a number of approaches that deal with the problem of physical symbol grounding in robotic systems from quite diverse perspectives. One of the notable instances of work that is exploiting encyclopaedic knowledge specifically for visual scene detection and interpretation, is the work by Pangercic et al.\textsuperscript{33}, where in the context of household robotic environments and working with a top-down guided 3D CAD model-based vision algorithm, they report results on how top-down guidance is influencing the perceptual process when exploiting how-to instructions which are extracted from linguistic descriptions from the web for the scenario of how to set the table for a meal. They mainly use Description Logics and OWL representations semantically based on the Cyc ontology. Their inspiration follows from some earlier work, focusing on practical and grounded knowledge representation systems for autonomous household robots\textsuperscript{40}, where they present a practical approach to robot knowledge representation that combines description logics knowledge bases with data mining and self observation modules so as to performing manipulation actions in a household environment. Already in this approach they combine the Cyc upper ontology with action related concepts that describe relations needed for mobile manipulation tasks. This is very interesting because we see that the strategy of attempting to integrate common-sense ontologies within the autonomous personal robots for the automated acquisition of grounded concepts through observations, is encouraged also in other works in the literature\textsuperscript{41}.

Despite the clear focus on the semantic aspect, the works mentioned above are
lacking a cooperative multi robot aspect, which we try to also capture in our current work. This specific aspect of transferring embodied concepts between perceptually heterogeneous robots has been studied in\(^\text{19}\), where the different perceptually heterogeneous embodiments communicate and effectively share knowledge that they have learned through individual experiences. Their approach is based on communication and knowledge exchange via social interaction, where properties of the world can be grounded differently in different robots. They use perceptual features to define properties of the objects, by methods that allow heterogeneous robots to learn grounded property representations (such as the colour categories), while they build models of object similarities and differences mapped to the respective representations. Their models are based on Gärdenfors’ conceptual spaces\(^\text{13}\) where object properties are learned and represented as regions in a metric space and are implemented via supervised learning of Gaussian Mixture Models. In the same context of multi-robot object localization, work by LeBlanc and Saffiotti\(^\text{24}\), addresses a novel approach for the problem of fusing information about object positions in multi-robot systems using using fuzzy logic. They use fuzzy sets to represent uncertain position information and fuzzy intersection to fuse this information while they evaluate their method using systematic experiments, which describe an input-error landscape for the performance of the approach.

As we see already the two groups of work mainly concern somewhat different aspects for the same problem, of collective perceptual information acquisition and exchange. The first one leaning on the semantic aspect while the second on the cooperative perceptual aspect. An instance of work by Mastrogiovanni et al., interestingly, considers a distributed architecture for symbolic data fusion which describes a truly distributed knowledge representation and data fusion system designed for highly integrated Ambient Intelligence applications\(^\text{30}\). Their architecture is based on the idea of an ecosystem of interacting artificial entities and essentially is a framework for collaborating agents to perform an intelligent multi-sensor data fusion. More specifically they focus on the cognitive layers leading the overall data acquisition process. Their approach has been tested in simulation, without investigating the integration between all the subsystems in a real, complex experimental set-up. This is what we primarily want to show through our evaluation. That by using current standard technologies and state-of-the-art algorithms and through the implementation of a sophisticated cooperative knowledge based perceptual anchoring system, it is possible to improve the cognitive abilities of the robots, regarding their communication, inference and planning skills.

In the approaches discussed so far, we see a lack of interaction with the human user. Initially work by Zender et al., describe an integrated robotic system for spatial understanding and situated interaction in indoor environments\(^\text{47}\). The functionalities of their system comprise of perception, natural language processing, learning and reasoning, all of them integrated into a mobile robot platform. Their work focuses on the principles behind the integration of the system, including cross-modal integration, ontology-based mediation and multiple levels of abstraction in percep-
tion. They are experimenting with their integrated CoSy Explorer system, based on the PECAS architecture which is a systematic combination of information fusion and continual planning techniques, between multiple concurrent robotic systems\textsuperscript{18}. Their CoSy Explorer/PECAS instantiation uses a hybrid spatial model combining SLAM, visual search and conceptual inference in the implemented scenarios.

Very closely related to our approach is also work by Kruijff et al. in situated dialogue and spatial organization where an HRI architecture for human-augmented mapping was used by the robot in order to augment its autonomously acquired metric map with qualitative information about locations and objects in the environment\textsuperscript{20}. They also focus on an ontology based approach to multi-layered conceptual spatial mapping that provides a common ground for human-robot dialogue by similarly combining acquired knowledge with innate conceptual common-sense knowledge in order to infer new knowledge. Their architecture bridges the gap between the rich semantic representations of the meaning expressed by verbal utterances on the one hand and the robot’s internal sensor-based world representation on the other.

7. Conclusion & Future Work

In real world applications, where the need for comprehensive human robot communication is increasing, one has implicitly to deal with grounding perceptual information to semantically structured common-sense concepts, whether this concerns either single or multi-robot systems. Knowledge should be expressed in a way that it is meaningful both for the robot and human. Many approaches to date tend to focus mostly on specific aspects in isolation and do not approach the physical symbol grounding problem. We acknowledge that all these aspects are quite important (including the grounding problem), therefore we mainly focus on the integration of the cooperative perceptual anchoring framework with knowledge representation techniques, to bring common-sense semantics to the architecture and enhance its communication skills (linguistically with the human and semantically with other robotic agents). For this integration we describe a model that defines different conceptual realms for information representation. We begin by acquiring and expressing perceptual information. Then we move to the representation of symbolically grounded information while finally we represent this information as perceptual knowledge in a common-sense knowledge base. We use this model in order to design and implement a cognitive perceptual system that consists of two heterogeneous agents, in the context of a smart home-like environment. Through the development of the system, we focus on the visual and spatial perceptual modalities and we give details on the corresponding functions. In the results we first see information about the performance characteristics of the system while then we discuss different ways which we can exploit the common-sense information for performing different communication tasks.

Due to the complexity of the system, but also the length of the current work,
there are certainly aspects, which we were not able to capture here. These include the algorithmic analysis of the anchoring functionalities, the grounding of other perceptual properties but also learning during the different stages of the anchoring process such as visual learning of objects without providing a-priori perceptual information (training) about the objects in the environment. Learning could also be explored in the grounding relations, where the perceptual agents do not rely on fixed grounding mechanisms, but rather on adaptive ones. Finally learning could also be exploited on the semantic level, where we teach the robotic agents new concepts linguistically, so that they ground and anchor the new concepts in their perceptual experiences. These learning scenarios can only be realized by having more advanced perceptual algorithms driving the visual perception. So in the future except of addressing the algorithmic aspects of the functionalities of anchoring, we aim to update the perceptual algorithms to support conceptual-perceptual information acquisition mechanisms, so as to train new objects autonomously while learning their grounding relations or conceptual common-sense information. At the same time it is quite important that we evaluate the scalability of the system where there are several agents (more than two) that cooperatively anchor and communicate information, eventually about a large number of objects.

Acknowledgments
This research project was funded by the Swedish Research Council - Vetenskapsrådet and whose support the authors gratefully acknowledge. Also we thank Cycorp Inc. for offering their ResearchCyc platform used in this research. Finally we thank Sahar Asadi and Mathias Broxvall for their generous help with this research.

References
REFERENCES


