Grounding the Interaction: Anchoring Situated Discourse in Everyday Human-Robot Interaction

Séverin Lemaignan, Raquel Ros, Emrah Akin Sisbot, Rachid Alami, Michael Beetz

International Journal of Social Robotics manuscript No.
(will be inserted by the editor)

Grounding the Interaction: Anchoring Situated Discourse in Everyday Human-Robot Interaction

Séverin Lemaignan · Raquel Ros · E. Akin Sisbot · Rachid Alami · Michael Beetz

the date of receipt and acceptance should be inserted later

Abstract This paper presents how extraction, representation and use of symbolic knowledge from real-world perception and human-robot verbal and non-verbal interaction can actually enable a grounded and shared model of the world that is suitable for later high-level tasks such as dialogue understanding. We show how the anchoring process itself relies on the situated nature of human-robot interactions. We present an integrated approach, including a specialized symbolic knowledge representation system based on Description Logics, and case studies on several robotic platforms that demonstrate these cognitive capabilities.

Fig. 1 Interacting with the robot in an everyday setup: the human asks for help in vague terms, the robot takes into account the human's spatial perspective to refine its understanding of the question.

1 Grounding Human Interaction Into Robot Knowledge

A messy table, covered with cardboard boxes, books, video tapes... Thomas is moving out and is packing everything with the help of Jido, his robot. "Jido, give me that", says Thomas while looking at a box that contains a video tape. The robot smoothly grasps the tape, and hands it to him.

While this kind of interaction (Fig. 1) should hopefully sound quite familiar in a foreseeable future, our robots are not yet quite up to the task, neither regarding natural language understanding nor plan-making and manipulation. To be combined together, those abilities require an unambiguous and shared representation of concepts (objects, agents, actions...) underlying the interaction: what are the prerequisites for such a sentence –"Jido, give me that"– to be understood by the robot, correctly interpreted in the spatial context of the interaction, and eventually transformed into an action?

The first step is to understand the meaning of the sentence. To this end, we must acquire the sentence, convert it into a useful syntactic form (probably through speech recognition), and understand the semantics of the sentence, *i.e.* What is referred by "*Jido*"? What is "*give*"? What is "*me*"? And "*that*"?

Working in a situated context, we need to *resolve* these semantics atoms, *i.e.* ground them in the sensory-motor space of the robot. For instance, "*that*" is a demonstrative pronoun that refers in this context to the object the human is focusing on.

Séverin Lemaignan · Raquel Ros · E. Akin Sisbot · Rachid Alami
CNRS - LAAS, 7 avenue du Colonel Roche, F-31077 Toulouse, France
Université de Toulouse, UPS, INSA, INP, ISAE, LAAS, F-31077 Toulouse, France
E-mail: {slemaign, rrosespi, sisbot, rachid}@laas.fr

Séverin Lemaignan · Michael Beetz
Intelligent Autonomous Systems, Technische Universität München, Munich, Germany
E-mail: {lemaigna, beetz}@in.tum.de

The next step is to extract and understand the *intended meaning* of the utterance as thought by the agent. In our example, Thomas obviously wants an action to be performed by the robot. The action parametrization is conveyed by the semantics attached to the words and the grammatical structure of the sentence. In our example, the type of action is given by the verb "*give*". Assuming the robot has some procedural knowledge attached to this symbol, the action type can be considered as grounded for the robot. We can as well understand that the recipient of the action is the human, the performer is the robot itself, and the object acted upon is the tape. The recipient, performer and object are three of the *thematic roles* [11] that qualify the *give* action. They are necessary to fully ground the sentence[1].

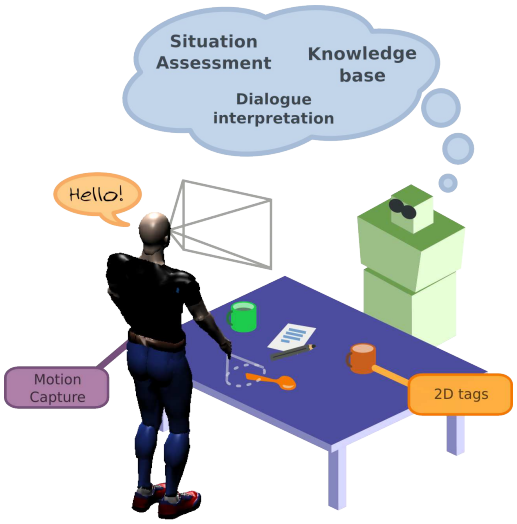


Fig. 2 Generic model of cognitive abilities interactions for grounding

1.1 Approach

In this work we propose an approach aiming at grounding the interaction with users (Figure 2 illustrates the general context). To this end, we have developed three distinct, inter-related cognitive components:

1) Physical environment modeling and *spatial reasoning* (grouped under the term *situation assessment*): this component is in charge of building and maintaining a coherent model of the physical world. This model is realistic in the sense that it relies on accurate 3D models of both manipulable objects and humans. It also has dedicated mechanisms to manage disappearing or occluded objects. The geometric model is used to compute several spatial properties of the scene that actually convert the original sensory data into symbolic beliefs, including relative locations of objects, visibility state, gestures (such as pointing), etc. Assuming that other agents are also represented in the model, the same computations are applied to analyze the scene from each agent's point of view (*i.e.* from their *perspectives*).

2) Knowledge representation and management: the robot is endowed with an active knowledge base that provides a symbolic model of its beliefs of the world, as well as models for each cognitive agent the robot interacts with. Used in combination with the situation assessment framework, this proves an essential feature ([31, 20]) to enable perspective-aware grounding of natural language.

Our platform relies on OWL-DL[2] ontologies and features continuous storage, querying and event triggering over the pool of facts known by the robot.

3) Dialogue processing: the third component includes natural language parsing capabilities, interactive disambiguation routines and concept anchoring. We focused our efforts on three classes of utterances commonly found in human-robot interaction: *statements* (*i.e.* new facts the human wants to inform the robot), *orders* (or more generically *desires*) and *questions on declarative knowledge* (whose answers do not require planning or prediction)[3].

1.2 Related Work

Our work builds on top of years of work in the artificial intelligence community around the symbol grounding issue: we can mention Searle [34] who introduces it with the *Chinese room* metaphor, Harnard [14] who coins the following (slightly paraphrased in the robotic context) definition of symbol grounding: *"make the meanings of the meaningless symbol tokens intrinsic to the robotic system"*, Coradeschi and Saffioti [7] who focus on robotics and define the meaning of *anchoring* as *"the process of creating and maintaining the correspondence between symbols and sensor data that refer to the same physical object"* and Ziemke [45] who elaborates on the need of embodiment for symbol grounding.

[1] This analysis has been inspired on the work of Austin et al. [2], where this type of sentences correspond to *speech acts*, comprising of *locutionary act* (the meaning), *illocutionary* (the intent) and possibly *perlocutionary acts* (implicit speaker's expectation).

[2] *Web Ontology Language - Description Logics*, a decidable subset of the first-order logics, `http://www.w3.org/TR/owl2-primer/`

[3] This would roughly cover the *representative* (sometimes referred as *assertive*) and *directive* illocutionary acts in Searle's [33] classification.

Our contribution relates to two narrower fields: natural language in embodied interaction contexts and knowledge acquisition and representation in robotic systems.

Processing natural language in situated contexts is already an established research field. In [31], Roy and Reiter summarize what they see as the main challenges to be tackled: cross-modal representation systems, association of words with perceptual and action categories, modeling of context, figuring out the right granularity of models, integrating temporal modeling and planning, ability to match past (learned) experiences with the current interaction and ability to take into account the human perspective. This list offers an interesting entry point to evaluate our contribution.

Kruijff et al. provides in [20] an up-to-date survey of literature on situated human-robot dialogue, focusing on formal representation systems, bi-directionality of the interaction and context building. They point out as well that compared to the cognitive psychology community, the "situated AI" community started only recently to take into account agents' focus of attention, perspective and temporal projection abilities.

Dialogue processing in real robots have been explored by several teams. Brick and Scheutz [6] have contributions regarding natural language processing in an incremental way, and how this enables instant backchannel feedback (like nodding). Hüwel et al. [16] propose the concept of *Situated Semantic Unit*: atoms are extracted from sentences exposing semantic links to other units. The parser tries to satisfy these links and rates the semantic interpretation of the sentence. Used in conjunction with ontologies, their approach offers robustness to ungrammatical or partial utterances. They validated the approach with an extensive user-study.

Zender et al. [44] address the generation of referring expressions (GRE [8]) in situated dialogue for topological knowledge. They consider both the reference resolution and reference description tasks, and rely on OWL-DL representation and SPARQL[4] to extract *topological contexts* from their knowledge base.

While mostly implemented on virtual agents, the GLAIR cognitive architecture by Shapiro and Bona [35] is an architecture explicitly built to tackle the grounding issue from the percept to the decision. It is a three-layers architecture: a *Knowledge Layer*, a low-level *Sensori-Actuator Layer* and an intermediate *Perceptuo-Motor Layer* that binds the previous two. The knowledge layer relies on a custom knowledge representation language (more expressive than first-order logic), and natural language processing capabilities similar to ours are available. The GLAIR project has been only demonstrated in a limited set of environments, but exhibits interesting features such as explicit management of contexts of facts and memory models (long term/short term, episodic/semantic).

Also worth mentioning, Mavridis and Roy [24] propose the idea of a *grounded situation model* which is an amodal model of the world where different sensing modalities, including verbal ones (the robot is able to *imagine* objects), are merged. Their framework also allows the management of the interaction history (the human can ask for a past event). They propose an implementation in an environment built on simple entities (a manipulator arm and colored balls).

In the field of symbolic knowledge processing for robots, Gunderson and Gunderson [12] introduce the concept of *reification* (based on both recognition and pre-afference) as an intermediate step between pattern recognition and symbol grounding. Their underlying storage of knowledge relies on ontologies and bio-inspired memory models. While sharing similar foundations to our work, their proposal is based on fairly simple perceptual modalities and does not develop complex symbolic models that could enable human-robot interaction.

Suh et al. [40] develop OMRKF, an ontology-based reasoning framework for robotics. They tackle the grounding problem by storing low-level facts (like SIFT visual features) in a layered symbolic architecture that works well in simple sensori-motor spaces. However this approach raises concerns regarding scalability and management of more complex entities or interactions.

Daoutis et al. [9] introduce one of the first complete architectures for grounded human-robot interaction. They successfully bind low-level percepts (including view-point independent SIFT based object recognition) to a high-level knowledge representation and reasoning system. They base their knowledge model directly on the *ResearchCyc* ontology (including the *MicroTheories* concept), used in combination with the CycL language. This enables second-order logic modeling and access to a large common-sense knowledge base.

Beetz et al. [4] proposes a cognitive architecture called CRAM (Cognitive Robot Abstract Machine) that integrates KnowRob [41], a knowledge processing framework based on Prolog. Its underlying storage is based on an OWL ontology, derived from OpenCyc. CRAM and KownRob have been demonstrated on several real-world scenarios, where natural language recipes extracted from Internet had to be translated into plans and executed in a kitchen environment, perceived and rebuilt on-line by the robots. While Prolog offers more flexible modeling (no constraints on the arity of predi-

[4] *SPARQL Protocol and RDF Query Language*, `http://www.w3.org/TR/rdf-sparql-query/`

cate, where Description Logics as used in our work are limited to binary predicates), it is based on the closed world assumption (if something cannot be inferred to be true, it is inferred to be false) whereas we rely on the open world assumption, which is more realistic in real world scenarios. A probabilistic extension of KnowRob, called ProbCog [17] is also available. While in principle possible, currently the CRAM architecture does not provide explicit support for interacting with humans.

1.3 Contributions

Besides proposing a new integration model for sensor data, natural language and symbolic knowledge repositories, our work extends these previous contributions by tackling more realistic human-robot interactions: less restricted speech understanding; ability to deal with complex, partially unknown human environments; and fully embodied (with arms, head,...) autonomous robots that manipulate a large range of household objects.

Three specific contributions are presented in this work: first, we introduce a versatile and light-weighted knowledge base that models in a formal framework, based on first-order logics, not only the robot's own beliefs but also every other cognitive agent the robot interacts with. This explicit modeling of other agents' belief states is used for the interaction and eases the implementation of various advanced cognitive behaviors like False-Beliefs [22] or interactive object discrimination.

Second, we have implemented a framework to extract symbolic facts from complex real scenes. It is based on a 3D model of the world that the robot builds on-line by merging different sensor modalities. It computes spatial relations between perceived objects in real-time and it allows for virtually *viewing* of the same scene from different points of view, enabling *visual* and *spatial agent perspective taking*.

Third, the same symbolic knowledge base enables richer language capabilities for the robot. We propose a new approach to natural language grounding that is robust, situated and more generic than what can be found in previous work on situated language grounding. We present several examples that include recognition and semantic validation of thematic roles or disambiguation based on attention foci.

Communcation between these components is build as streams of symbolic facts, where knowledge manipulated by the robot is made explicit. This leads us to the idea of a *knowledge-oriented architecture*, which is discussed at the end of the article.

These points highlight some original aspects of a larger cognitive architecture that has been deployed and tested on several mobile robotic platforms (including both humanoid robots and service robots), demonstrating the versatility and hardware-agnosticism of these developments.

In the next section we present the ORO (*Open-Robots Ontology* server) knowledge base. We present it first, along with its objectives, since it is the knowledge *hub* of the system, used pervasively by other components. Section 3 presents SPARK (for *SPAtial Reasoning & Knowledge*), the component that merges perceptual information with a coherent geometric model and builds a symbolic interpretation of the world from the robot's point of view, as well as an individual symbolic model for each agent currently present in the environment. Section 4 covers in detail the dialogue processing component and its relation with ORO. Section 5 presents three use-cases that were conducted on two different robotic platforms: the *Naming* experiment, where a robot anchors new knowledge in its model through verbal interaction; the *Spy Game*, where either the user or the robot tries to guess which object the other player is thinking of; and a more complex experiment situated in an everyday setup, where the robot builds models for several agents and interacts with the users using this knowledge. Finally, Section 6 concludes and discusses the work presented in this paper.

2 ORO, a Knowledge Management Platform

2.1 On Knowledge Representation

While *knowledge* has no general definition that researchers agree on, for our own purposes we define knowledge as *information interpreted in the cultural and social context of the robot*, where information is a *statement* or an *assertion* about the world[5]. In practical terms, knowledge is made of statements that are *contextualized*, if possible *synthesized*, and *limited* to a domain of validity. These three features have important consequences for the way a knowledge representation and storage system must be designed. Let us examine them:

Contextualizing is the ability for a cognitive system to connect a fact with a *cultural context*, an *interpretive scope* and the set of other facts previously acquired by the agent.

We call *cultural context* a broad set of common, general facts that are considered widely accepted among

[5] In this paper, statements are always triples ⟨`subject predicate object`⟩, *i.e.* binary relations between entities.

the interactors (*e.g.* "bottles may contain water"). This knowledge is often referred as *common-sense knowledge*.

By *interpretive scope* we mean that a concept may have different interpretations depending on the agent, the current situation or the time frame the statement belongs to. Since a fact in one scope can be different (or even inconsistent) with a fact in another scope (for instance, one object can be visible for the robot and invisible for another agent), the underlying knowledge representation system must properly handle these interpretive frameworks.

Note that the focus of ORO is on enabling such context to be effectively represented rather than actually identifying the current context. While several approaches for building contextualized knowledge are proposed in this paper (symbolic environment interpretation, perspective taking, grounded natural language resolution, self-awareness of its own activity), much remains to be done for a robot to actually identify its current context as well as contexts that may be referred to.

Synthesis corresponds to the identification of facts and their components (concepts and predicates) with respect to other facts. For instance, if the robot observes a human sitting down at a table, and at the same time, we tell it that "Peter is sitting at the table", we would like the robot to infer that "Peter" may be the name of the human. *Synthesis* refers to the fact that several, *a priori* uncorrelated, facts must be associated with the same common concept. This process requires the ability to control the logical consistency of the knowledge corpus. To continue with the previous example, if we add the fact that the human that is sitting is a woman, the synthesis "Peter is the name of the human" is not valid anymore.

Domain of validity specifies the scope in which information is (believed to be) true. It covers several aspects: temporal, situational and probabilistic. While related to the previous concept of *interpretive scopes*, the domain of validity addresses the question whether a fact must be or not considered in a given context. This validity limitation is not usually carried by the fact itself. In the previous example, for instance, the robot observes a human sitting at a table. The fact "a human is sitting at the table" is true only for a limited period of time, until the human stands up. This period of time is not directly accessible (the robot does not know how long the human plans to stay), but the knowledge representation must be able to deal with this uncertainty and should explicitly label this fact as being limited in time.

These three aspects lead us to envisage a knowledge representation system characterized by the following abilities:

- represent raw information,
- render a general cultural background, in the form of common-sense knowledge,
- attach interpretive scopes to new statements,
- add and connect new statements to knowledge already present,
- store restrictions on the domain of validity of the knowledge.

Besides, the following active processes would be desirable:

- acquire and maintain knowledge perceived from the physical world or retrieved from other sources (interaction with other agents, web-based contents,...)
- synthesize facts as much as possible,
- monitor contexts and accordingly manage the validity of the stored knowledge,
- ensure the logical consistency of the knowledge repository, and explicit inconsistencies when required[6].

This list does not cover all the possible features that could be exposed by a symbolic knowledge management system. Bio-inspired memory management (the ability to forget or reinforce knowledge) or curiosity (the ability to identify lacking knowledge and actively trigger behaviours to acquire it –Hawes et al. [15] have contributed in this area with *Dora*, a robot endowed with motivation mechanisms to explore unknown regions of the environment–), to give some examples, could arguably be added to the list. However, this first analysis sets a convenient reference frame to understand and evaluate knowledge representation systems, including the ORO knowledge management system we propose.

2.2 ORO Architecture

The ORO platform [21] is primarily designed as a central knowledge storage service implemented as a server where the robot components can add or query statements at run-time. Figure 3 illustrates the overall architecture. The *front-end* accepts and manages connections from client components. The clients' requests are processed by a set of internal modules: basic operations on statements, but also higher cognitive and human-robot interaction related functionalities are available

[6] One may argue that the real world is inherently inconsistent. In this article, we make a *consistent world* assumption, in order to leverage reasoning capabilities of the first-order logics. This is supported by the natural tendency of humans themselves to provide a consistent explanation of their world.

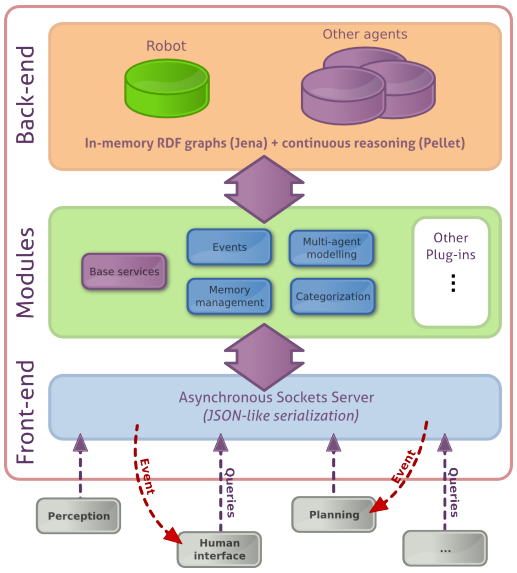


Fig. 3 Overview of the ORO architecture.

(detailed thereafter). External plugins can also be easily added. The modules rely on several parallel ontology *back-ends* where the knowledge is actually stored.

Knowledge is represented in ORO in the Description Logics formalism (using the OWL-DL – *Web Ontology Language - Description Logics*– language), as RDF (*Resource Description Framework*) triples (for instance ⟨`robot isIn kitchen`⟩). We use the Jena[7] framework as the underlying library to load and build an in-memory OWL model. We use it in conjunction with the Pellet[8] reasoner to ensure the continuous classification of the OWL concept graph: at run-time, newly added statements are continuously reasoned about (*classified*), and at any time, the server exposes a complete set of statements, both the asserted ones and the inferred ones. For instance, if ⟨`socrates type Human`⟩ and ⟨`Human subClassOf Mortal`⟩ are asserted, the server transparently adds the inferred statement ⟨`socrates type Mortal`⟩, and a later query retrieving all mortal entities would return `socrates` as well. The language of the OWL family make the *Open World Assumption* (if a fact can not be inferred as true, it does not mean that it is inferred to be false), and the Pellet reasoner honors this assumption as well.

2.3 The OpenRobots Common-Sense Ontology

The first requirement identified in section 2.1 refers to the modeling of a cultural background, a common-sense knowledge assumed to be shared by all agents. The ORO server can be loaded with an initial set of statements which we call the *OpenRobots Common Sense Ontology*. It defines a small set of concepts (and implicitly, a vocabulary) that can be used by all the modules of the robot to unambiguously add or query facts. Moreover, the same ontology declares rules and logical properties that are later on used for inference.

The *OpenRobots Common Sense Ontology* defines a small set of classes (56 are currently defined) and predicates (60 are currently defined) focused on concepts useful for human-robot interaction. It includes both very broad categories like `SpatialThing`, `Event` or `Action`, and much more concrete concepts as `Table`, `Book` or colors. Available predicates allow us to describe the state of the agents and the world with relations like `isOn`, `sees`, `currentlyPerforms`, etc.

Several significant projects are trying to provide such a machine-processable repository of common sense facts produced by humans (the OpenMind project[9], for instance). These knowledge bases are valuable but remain difficult to use in a pervasive way because of both their incompleteness and the lack of good connections with underlying, unambiguous concepts.

Our common sense ontology is closely aligned with the open-source OpenCyc[10] upper ontology. OpenCyc defines a large taxonomy of concepts and semantic relationships between concepts that are used in several other projects (WordNet, DBpedia). This potentially eases the exchange and addition of knowledge from these other sources. Moreover, it also enables knowledge exchange with other robots (for instance, the works previously mentioned by Daoutis and Tenorth rely on the same Cyc concepts).

2.4 Reasoning and Dynamic Knowledge Structuring

As previously mentioned, ontologies in ORO are written in OWL. The Pellet reasoner supports most of the OWL constructs and allows several types of reasoning:

- inheritance
- property axioms
  - entailments based on predicates' domain and range,
  - cardinality constraints (including `allValue`, `someValue`, `hasValue`),
  - property characteristics (symmetry, transitivity)
- class restrictions like:
  `Bottle` ≡ `Artifact` **that** (`hasShape` **value** `cylinderShape`)[11]
- set operations like:
  `Color` ≡ **unionOf**(`blue`, `green`, `orange`, `black`...)

[7] `http://jena.sourceforge.net/`
[8] `http://clarkparsia.com/pellet/`
[9] `http://www.openmind.org/`
[10] `http://www.opencyc.org`
[11] This example uses the *Manchester syntax*, `http://www.w3.org/TR/owl2-manchester-syntax/`

– generic SWRL (*Semantic Web Rule Language*) rules like:
`looksAt(?agt, ?obj)` ∧ `pointsAt(?agt,?obj)`
⇒ `focusesOn(?agt, ?obj)`

We provide in ORO accessors to query, add or remove all these properties and restrictions (except the SWRL rules) at run-time. This allows knowledge introspection and enables the robot to alter its own knowledge structures (the so-called *T-Box* model) during its life-time by adding new constraints and properties to classes and predicates. The *Naming* experiment (section 5.1) gives a simple example of such knowledge restructuring.

2.5 ORO Features

Besides storing and reasoning about knowledge, we have developed in ORO several features to manage knowledge at higher level:

2.5.1 Base Functionalities

ORO offers an extended set of methods to process facts at the triples level, including:

– statement (*i.e.* RDF triples) insertion, removal, update,
– pattern-based statements removal,
– pattern-based queries (for instance, ⟨ `* isOn table`⟩, which means "return me all objects on table") and filters (for instance, `weight < 150.0`),
– consistency check, insertion of statements with consistency constraint (only if the new fact does not lead to inconsistencies),
– fast concept lookup, with possible multi-lingual support (through the `@lang` XML annotation, labels of concept can be translated, and specific translations can be queried for),
– standard SPARQL queries.

While these basic functionalities enable the incremental construction (and exploitation) of a consistent knowledge model, the *Common Sense Ontology* helps build assertions that are related to previous ones by offering a predefined vocabulary.

2.5.2 Representation of Alternative Cognitive Models

As pictured in Figure 3, ORO stores independent cognitive models for each agent it interacts with. When ORO actually identifies a new agent (or infers that some instance is an agent), it automatically creates a new, separate, in-memory OWL model for that agent. Thus, different robot components, like supervision or situation assessment, may then store the agents' beliefs in separate models. All knowledge processing functions in the robot's primary model are equally available in every agent's model, which allows us to store and reason on different (and possibly globally inconsistent) models of the world.

Each of these models is independent and logically consistent, enabling reasoning on different perspectives of the world that would otherwise be considered as globally inconsistent (for instance, an object can be visible for the robot but not for the human. This object can have at the same time the property `isVisible true` and `isVisible false` in two different models).

This feature actually allows us to consider the robot to be endowed with a *theory of mind* [32]: the robot can explicitly model the belief state of its interactors, opening new possibilities for the control architecture. In section 5.3.2 we present an example of how we use this feature to make sense of user sentences from his/her point of view. Moreover, these multiple models can be viewed as different interpretive scopes, allowing the robot to interpret the same reality from different points of view.

2.5.3 Categorization

We have implemented several algorithms (common ancestors, computation of the best discriminant [29]) to help the robot cluster a set of concepts based on their symbolic similarities. One particular application of these functions is discrimination. While interacting with a user, the robot quite often needs to clarify an ambiguity produced by its human partner. For instance, a user may refer to a "bottle" where two bottles are currently visible. Discrimination routines can identify possible (symbolic) differences (*e.g.* the color or the size of the bottles) that permit the robot to ask an accurate question to the user in order to solve the ambiguity. This discrimination can occur from the robot's perspective or from a specific agent's perspective. Usage of these categorization abilities are illustrated in Sections 4.3 and 5.2.

2.5.4 Memory Profiles

We have designed a simplified bio-inspired memory model to store statements in different *memory profiles*. These include *short term memory* and *long term memory*. Each profile is characterized with a lifetime, which is assigned to the stored facts. When the lifetime of a fact expires, ORO automatically removes it.

2.5.5 The Events Framework

Lastly, ORO allows external modules to be triggered when specific events occur. For instance, when a logical sentence becomes true or false, or if a new instance of a certain class is added. One immediate application is reactive supervision: a component could for instance subscribe to events of kind [`?agent isVisible true`, `?agent type Human`]. As soon as the perception layer detects a human in the robot's field of view and accordingly updates the knowledge base, the supervision component would be triggered back. The event framework also takes advantage of the inference capabilities of ORO. Thus an event can be indirectly triggered if its triggering conditions can be inferred to be true.

The next sections describe how symbolic knowledge is actually produced and added to ORO.

3 Geometric Reasoner for Situation Assessment

Anchoring perceptions in a symbolic model requires perception abilities and their symbolic interpretation. In this section we present SPARK (*SPAtial Reasoning & Knowledge* [36]), a situation assessment reasoner that generates relevant symbolic information from the geometry of the environment with respect to relations between objects, robots and humans. Moreover, the notion of *Perspective Taking* [10, 42] is employed at the heart of the reasoner to provide the robot with the ability to put itself at the human's place and to reason about the world from different perspectives.

As mentioned in the introduction, this paper does not focus on the sensor-level perception. We rather assume that the perception of the humans and the objects is provided as a list of unique identifiers with associated 3D meshes and 6DOF poses.

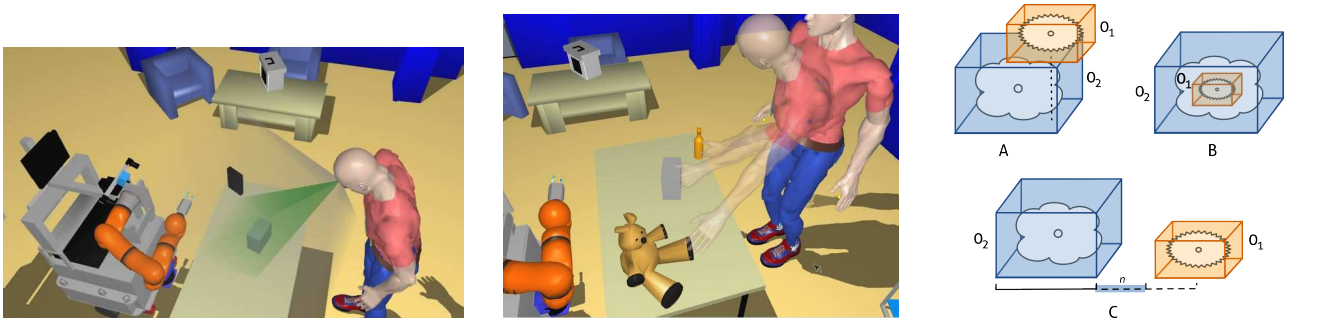
3.1 Capabilities

There are a number of common properties for a robot and a human related to their capabilities in a given situation: they can both reach, grasp, look at, point at, etc. In our context, we group robots and humans into a single category. Thus, we define agents as entities that can act in the environment and manipulate it. In this work we focus on the following capabilities from each agent's perspective[12]:

- *Sees*: An important ability to know about an agent is to predict "what it can see", *i.e.* what is within its field of view (FOV). A robot being able to compute this information can then act accordingly. An example would be a clarification scenario where the human is searching for an object and the robot is able to infer that he/she is looking for the one that is not visible (otherwise the user would not be searching for it). In Figure 4*a* the field of view of a person is illustrated with a grey cone (broader one). While he is able to see the two small boxes on the table in front of him, the big box on his right is out of his FOV, and therefore, he is not able to see it.
- *Looks At*: this relation corresponds to what the agent is focused on, *i.e.* where its focus of attention is directed. This model is based on a narrower field of view, the field of attention (FOA). Figure 4*a* shows the field of attention of a person with a green cone (narrower one). In this example only the grey box satisfies the `looksAt` relation.
- *Points At*: verifies whether an object is pointed at by an agent. This relation is particularly useful during interaction when one of the agents is referring to an object saying "this" or "that" while pointing at it. Section 5.3.3 describes in more detail the combination of both sources of information (verbal and non-verbal).
  If a big object occludes a smaller one, and an agent is pointing at them, the outcome of the evaluation will result only in one relation, *i.e.* ⟨`agent_01 pointsAt object_01`⟩ since the small one is not visible to the agent. On the contrary, if the small object is in front of the big one, then both objects will satisfy the relation, which may generate an ambiguity (which object the agent refers to?) that should be solved through higher level reasoning (*e.g.* context analysis or clarification through verbal interaction).
- *Reachable*: it allows the robot to estimate the agent's capability to reach an object, which is fundamental for task planning. For example, if the user asks the robot to give him/her an object, the robot must compute a transfer point where the user is able to get the object afterward. Figure 4*b* shows different reachability postures for each object on the table. In the example, the bottle and the box are both reachable for the human, but the teddy bear is too far. Instead, from the robot's perspective, the teddy bear is reachable for it, while the bottle is not.

While the first three relations (`sees`, `looksAt` and `pointsAt`) are computed through a model based approach, the latter one is based on the Generalized Inverse Kinematics with pseudo inverse method [26, 3] to find a collision free posture for the agent where its end-effector is at the center of the object within a given tolerance.

[12] Note that each of the capabilities described are computed from each agent point of view, and therefore, also stored in different models in ORO for further use at the decisional level.



(a) (b) (c)

Fig. 4 (a) Field of view (FOV) and the field of attention (FOA) of the human. (b) Different reaching postures for the human. (c) Spatial relations between two objects: A) `isOn` relation, B) `isIn` relation, and C) `isNextTo` relation.

The details of these computations are out of the scope of this article.

3.2 Locations

One way of referring to object's positions is based on human's symbolic descriptors, instead of using their precise position. In fact, in many cases, this information is the most precise information available since humans do not store the numeric coordinates of objects. These type of descriptors have been studied in the context of language grounding ([27, 23, 28, 18, 5]). In this work we focus on the following relations which are computed with respect to the position of the agents and the objects:

- *Location according to an agent*: The predicate `isLocatedAt` represents spatial locations between agents and objects. For example we say "it is on my right, on your left, ..." We compute these spatial locations by dividing the space around the referent (an agent) into n regions based on arbitrary angle values relative to the referent orientation. For example, for $n = 4$ we would have the space divided into *front, left, right* and *back*. Additionally, two proximity values, *near* and *far*, may also be considered. The number of regions and proximity values can be chosen depending on the context where the interaction takes place.
- *Location according to an object*: We can also refer to object locations with respect to other objects in the environment, such as *above, next to, in*, etc. In this work we compute three main relations based on the bounding box and center of mass of the objects (Figure 4*c*):
  - `isOn`: computes if an object O_1 is on another object O_2 by evaluating the center of mass of O_1 according to the bounding box of O_2.
  - `isIn`: evaluates if an object O_1 is inside another object O_2 based on their bounding boxes BB_{O_1} and BB_{O_2}.
  - `isNextTo`: indicates whether an object O_1 is next to another object O_2. We cannot use a simple distance threshold to determine if two objects are next to each other since the relation is highly dependent on the dimensions of the objects. For instance, the maximum distance between large objects (*e.g.* two houses) to consider them as being next to each other is much larger than the maximum distance we would consider for two small objects (*e.g.* two bottles). Thus, the relation between the dimensions and the distances of the objects are taken into account.

To ensure the different agent models are up-to-date, all these properties are always computed on-line, each time the current state of the world changes.

SPARK can be compared to the *Grounded Situation Model* (GSM) introduced by Mavridis and Roy [24] in the sense that they both provide an amodal physical representation of the world used as a mediator between the sensor space and symbolic models. They have however different features: while GSM enables representation of time and imaginary objects (whose existence is hinted by verbal assertions from a human, also called *presupposition accomodation*), SPARK offers a richer 3D model that enables the computation of several spatial relationships between objects and an effective implementation of perspective taking capabilities.

4 The Natural Language Grounding Process

Verbal interaction with human presents two categories of challenges: syntactic ones, and semantic ones. The robot must be able to process and analyze the structure of human utterances, *i.e.* natural language sentences,

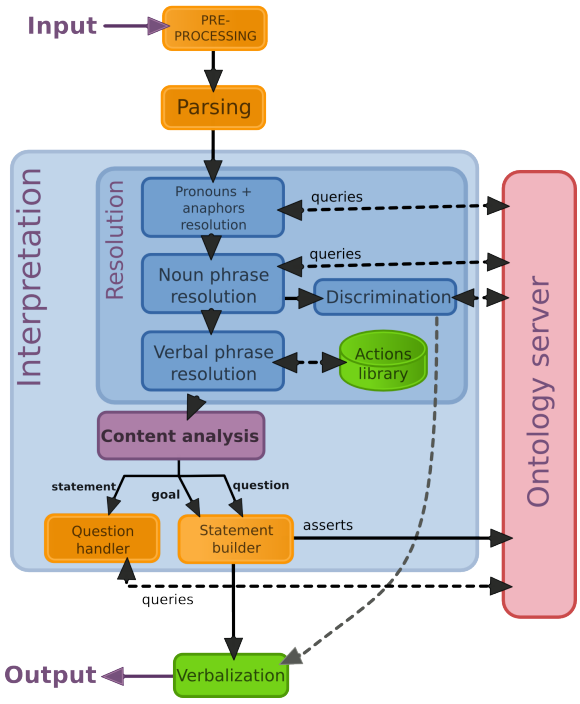


Fig. 5 The DIALOGS module has three main steps: the parsing, the interpretation and the verbalization. The interpretation module is responsible for both the *resolution* and the semantic content *analysis and translation*.

and then make sense of them. As stated in the introduction, we process three categories of sentences: *statements*, *desires* and *questions* that can be answered from the declarative knowledge present in the robot knowledge base (a choice similar to the *Behaviour Cycle* in the GLAIR architecture [35]). In our work, the grounding process of the human discourse consists in extracting either the *informational* content of the sentence to produce statements or its *intentional* content (*i.e.* performative value) to collect orders and questions. We do not claim any contribution to the field of computational linguists (see [20] for a survey of formal approaches to natural language processing in the robotics field). Our main contribution here is the grounding (we call it *resolution*) of concepts involved in the human discourse through the robot's own knowledge.

To this end we have developed a dedicated module called DIALOGS that processes human input in natural language, grounds the concepts in the robot's knowledge and eventually translates the discourse in a set of queries or declarative OWL/RDF statements. Figure 5 shows the DIALOGS module architecture. The user's input is first pre-processed. For instance, *I'm* constructs are expanded into *I am* and then parsed. The parser is a custom-made, rule-based (*i.e.* grammar-free) tool that extracts the grammatical structure from the user's

```
>> IMPERATIVE
VP: remember (present simple)
    SUBSENTENCE (aim: that)
      NP: I
      VP: want (present simple)
        direct objects:
          NP: you
        secondary VP: give ()
              direct objects:
                NP: my nice blue bottle
              indirect objects:
                NP: me
```

Fig. 6 Raw output of the DIALOGS parser after processing the sentence: "remember that I want you to give me my nice blue bottle." Nominal groups are not grounded yet.

sentence. Figure 6 shows an example of the raw output of the parser for a moderately complex sentence.

The output of the parser is then sent to the *interpretation* module, the core of the component. Interpretation consists in three distinct operations: the sentence *resolution* (concepts grounding), the *content analysis* (what is the intent of the utterance: information, question or desire) and the *statement building* (translation into RDF statements).

The sentence resolution has three steps: *(i)* pronouns and anaphora are replaced by the correct speaker ID and the ID of the last object referred to (extracted from the dialogue history) respectively, *(ii)* nominal groups are disambiguated and grounded (noun phrase resolution), and *(iii)* verbal groups are resolved and their associated *thematic roles* are retrieved (verb phrase resolution).

Algorithm 4.1: RESOLUTION(*sentence*, *currentSpeaker*)

$\mathcal{G} \leftarrow$ PARSENOMINALGROUPS(*sentence*)
for each $g \in \mathcal{G}$
do
- $\mathcal{D} \leftarrow$ GENERATEDESCRIPTION(g) (1)
- *candidates* $\leftarrow$ ONTOLOGY.FIND($\mathcal{D}$) (2)
- **if** $|candidates| = 0$
  - **then** **output** (Couldn't resolve the group!); **exit**
  - **else if** $|candidates| = 1$ **then** $id \leftarrow candidates[0]$
  - **else**
    - **if** ONTOLOGY.CHECKEQUIVALENT(*candidates*) **then** $id \leftarrow candidates[0]$
    - **else** $id \leftarrow$ DISCRIMINATION(*candidates*) (3)
- REPLACE(g, id, *sentence*)

As represented in Figure 5, interpretation tightly relies on the communication with the knowledge base. All the concepts the robot manipulates are stored in the ontology server and retrieved through logical queries, except for the verbs that are currently stored in a dedicated library (the *action library* in the diagram).

In order to better understand the overall process of the DIALOGS module and its relation with ORO, we

Initial knowledge human_01	*Human input*
⟨banana_01 type Banana⟩ ⟨banana_01 hasColor yellow⟩	"The yellow banana is big!"

Generated partial statements	*Newly created statements*
⟨?obj type Banana⟩ ⟨?obj hasColor yellow⟩ ⇒ ?obj = banana_01	⟨banana_01 hasSize big⟩

Fig. 7 First example: content extraction. "⇒" represents the output of the ontology server.

Initial knowledge human_01	*Human input*
⟨banana_01 type Banana⟩ ⟨banana_01 hasColor yellow⟩	"Give me the banana."

Generated partial statements	*Newly created statements*
⟨?obj type Banana⟩ ⇒ ?obj = banana_01	⟨human_01 desires sit_a3⟩ ⟨sit_a3 performedBy myself⟩ ⟨sit_a3 actsOnObject banana_01⟩ ⟨sit_a3 receivedBy human_01⟩

Fig. 8 Second example: processing an order.

next describe the different steps of the approach based on three examples. In these examples we assume that some initial facts are present in the knowledge base (section 6.2.1 discusses how the initial knowledge can be acquired), both in the robot's own model and in the human's model. Since the robot tries to ground a human utterance, all queries are sent to the human model in order to interpret it from the human perspective.

4.1 Informational Content Extraction

Figure 7 shows a first example of human discourse grounding and the extraction of informational content. We assume that the robot knowledge base only contains two initial statements in the human model. The user asserts a new one: "The yellow banana is big!". We first want to match the nominal group *The yellow banana* to an already known concept (algorithm 4.1), and second to translate the property *is big* into a predicate (`hasSize`) to state its semantics.

To resolve the nominal group *The yellow banana* a set of partial statements that describe the concept is generated based on the grammatical parsing of the sentence (algorithm 4.1(1)). The parsed tree of each nominal group is translated into statements based on a set of rules. In the example, a banana (⟨?obj type Banana⟩) that is yellow (⟨?obj hasColor yellow⟩)[13]. Based on these partial statements a SPARQL query is sent to the ontology server to retrieve possible instances that match the description (algorithm 4.1(2)).

In this first simple case, the concept `banana_01` is unambiguously matched (since there is only one possible banana) and returned. Finally, we can now add the new information provided by the human, *i.e.* the new statement ⟨banana_01 hasSize big⟩, to the human model in the ontology server.

[13] Predicates like `hasColor` or `hasSize` that bind `banana_01` to adjectives are extracted from a predefined database of [*Predicate* → *AdjectiveCategory*], and falls back on the generic `hasFeature` predicate if the adjective is not known.

4.2 Intentional Content Through Verb Resolution

The sentence in the first example is built with the state verb *be* at indicative. Let us examine a different example with an action verb at imperative mode (an order): "Give me the banana". The process is described in Figure 8.

In order to capture the intentional content of a sentence (for example, an order) we need to retain the semantics of the verb and its complements. *Thematic roles* allow for semantically linking a verb to its complements. There is no general agreement amongst linguists on a comprehensive list of thematic roles. The amount and the granularity of roles varies a lot in the literature [13]. We thus use a small set of them, which matches the relations the robot can actually achieve (we discuss possible extensions in the conclusion). For instance, in the second example, the verb *give* has three thematic roles: `performedBy`, `actsOnObject` and `receivedBy`.

The list of actions the robot can plan for (currently *take*, *place*, *give*, *show*, *hide* and *move*) along with possible synonyms (for example, *to pick* is set as a synonym of *to take*) and their associated thematic roles are stored in a predefined library of actions. For each action we identify and store: the role of the subject in the sentence (always `performedBy`); the role of the direct object (for instance, `actsOnObject`); and the role of each of the indirect objects with their optional prepositions (for instance, `receivedBy`)[14]. Moreover, through the ontology we check that each holder of a role is semantically consistent. For instance, the action *Give* must have a manipulable physical item (`Artifact`) as direct object. Thus, if the concept the robot finds for the thematic role `actsOnObject` cannot be inferred to be an artifact, the robot goes back to the human saying it does not understand.

This second example also shows the pronoun reference resolution: "me" is replaced by the id of the current speaker, while "you" is replaced by `myself` (`myself` always

[14] Note that in example 2, "give me the banana", the pronoun "me" appears before "banana", while it is an indirect complement — "give it **to me**". The parser correctly handles these cases.

Initial knowledge model of `human_01`

⟨`banana_01 type Banana`⟩
⟨`banana_01 hasColor yellow`⟩
⟨`banana_02 type Banana`⟩
⟨`banana_02 hasColor green`⟩

Human input

"The banana is good."

Generated partial statements

⟨`?obj type Banana`⟩
⇒ `?obj = [banana_01, banana_02]`

Discrimination process

`discriminate([banana_01, banana_02])`
⇒ `?hasColor = [yellow, green]`

Robot output speech

"The yellow one or the green one?"

Human answer

"The green one."

Extended human input

"The green banana is good."

Generated partial statements

⟨`?obj type Banana`⟩
⟨`?obj hasColor green`⟩
⇒ `?obj = [banana_02]`

Newly created statements

⟨`banana_02 hasFeature good`⟩

Fig. 9 Ambiguity resolution: in this example, "banana" can refer to the yellow banana (`banana_01`) or the green one (`banana_02`). Discrimination routines handle the disambiguation process.

represents the robot itself). When present, anaphoras (references to previous concepts like "give me the banana, I like **it**.") are also resolved in the same step.

Once the sentence is completely resolved and translated into a formal representation (a human desire in this example[15]), we store it in the ontology server. The robot's decisional/executive layers can then decide whether to execute the order or not.

4.3 Informational Content Extraction Requiring Clarification

This last example (Figure 9) shows the resolution of ambiguous concepts. In this case the user refers to "the banana" while two instances of the `Banana` class exist in the ontology. The robot needs to find out to which instance the user is actually referring to. To this end, disambiguation routines (algorithm 4.1(3), see [29] for details of the routines) find differences between the instances (in the example, one banana is yellow while the other one is green) and build a sentence through the *verbalization* module to ask the user a closed question that will help clarify the ambiguity: "Is it yellow or green?" The user's answer is parsed and merged with the previous sentence. The resulting, augmented, sentence ("The green banana is good") goes again through all the interpretation steps. This process is repeated until no ambiguities arise. In the example, the `banana_02` is finally returned.

Several other strategies are used in parallel to disambiguate concepts without having to ask for more information to the human:

- Which objects are currently visible to the human? If only one of them, then it is probably the one the user is talking about.
- Did a previous interaction involved a specific object that would still be the subject of the current sentence?
- Is the user looking or pointing at a specific object?

While no examples involving questions have been detailed, factual *wh-* questions and polar (*yes/no*) questions can be processed in a similar way by DIALOGS. For instance, a question like "What is on the table?" is grounded (to extract the relation `isOn` and to find what *table* refers to) and transformed into the following kind of query: `find ?var [`⟨`?var isOn table1`⟩`]`. Answers are converted back to a full sentence by the *verbalization* module, and uttered to the human.

5 Experiments

This section presents several experiments where the different features of our approach are represented. The experiments have been conducted on two different platforms: the *Rosie* manipulator from the Technical Universtiy of Munich, a dual-arm, holonomic service robot, running the ROS[16] middleware; and the *Jido* robot [1] from LAAS-CNRS, a similar, single-arm, service robot, running the LAAS's GENOM/POCOLIBS stack. Both platforms share the use of ORO, the ontology server described in this work.

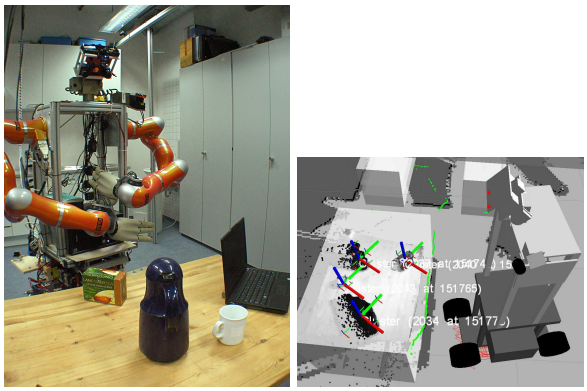
5.1 *Naming* Experiment

The *Naming* task uses ORO to anchor perception into the robot's knowledge through interaction with the user. This task has been implemented on the Rosie robot at TU Munich.

The robot selects an unknown object from the table, shows it to the user, and asks about its name and type

[15] Orders are here represented as human desires: the human desires a specific new situation.

[16] *Robotic Operating System*, `http://www.ros.org`



(a) (b)

Fig. 10 (a) Rosie, looking for objects it may know, and (b) view from Willow Garage's RViz. The clusters of point are given a unique identifier by the perception module. This allows the supervision module to create the link between the physical objects and their symbolic representation in ORO.

rosie	Servus! The Bavarian robot is up and fit!
rosie	What is the name of the white cluster on the table?
human	"Rosie's mug"
rosie	*[*⟨`obj_01 label "Rosie's mug"`⟩ *added]*
	What kind of object is "Rosie's mug"?
human	Mug
rosie	*[*⟨`obj_01 type Mug`⟩ *added]*
	What is a Mug?
human	Tableware
rosie	*[*⟨`Mug subClassOf Tableware`⟩ *added. Rosie already knows about* Tableware *(it belongs to the common-sense ontology), so the description of this object ends.]*
	(...)
rosie	What is the name of the black cluster on the table?
human	"my_laptop"
rosie	What kind of object is "my_laptop"?
human	Computer
rosie	What is a Computer?
human	Artifact
rosie	Do you want me to learn more objects?
human	no
rosie	These objects are odd: "my_laptop"

Fig. 11 Transcript of a Jabber session with the robot Rosie.

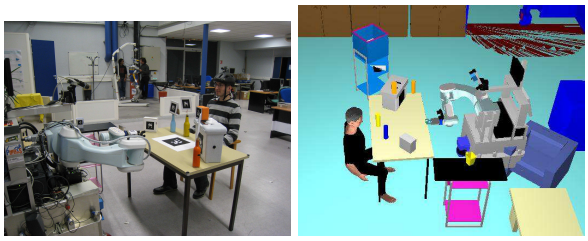
(Figure 10). The user interacts with the robot through a dedicated bridge, allowing him/her to chat with the robot with a standard Jabber instant messaging client. Figure 11 shows a chat session with Rosie performing the naming experiment. The human describes the object until the robot recognizes a type it already knows. More precisely, the robot recursively asks the human for the categories the object belongs to until reaching a known one. At this point, the robot is able to connect the new object information to already known concepts. Therefore, the robot accurately anchors perception in its symbolic model and it is able to reason about it. The learning process starts over again with each unknown object. To further exploit the knowledge reasoner in this experiment, we ask the robot which object does not belong to a typical breakfast table, *i.e.* we ask for the odd objects. In this case, it corresponds to objects that are neither of type `Tableware` nor `Food`.

The supervision module (CRAM[17] [4]) automatically updates the ORO server whenever an object enters or leaves the field of view. Therefore, the integration of ORO can be seen as an extension of the robot's belief state that not only contains abstract identifiers of the internal object representation used in plans, but also the semantics and roles of objects in the scenario.

5.2 *Spy Game* Experiment

This game is based on the traditional children game "I Spy". The idea is to discover the object or concept

[17] CRAM (Cognitive Robotic Abstract Machine) is a RPL-derived framework for rapid development of cognitive robot control programs we currently develop.



(a) (b)

Fig. 12 Spy game scenario: (a) Real environment and (b) 3D environment model, viewed in SPARK.

one of the participants is thinking of by asking questions such as: "Is it green? Is it a machine? Is it on your left?", etc. When playing, children exploit their knowledge about the world while categorizing and describing objects through useful discriminants that will allow them to find out the answer as fast as possible while including perspective taking abilities [25].

The scenario for this game (Figure 12) consists on a face-to-face interaction where the human thinks of an object present in the environment, while the robot queries the human until either discovering the object or giving up [30]. The game starts with the human user giving a first hint (communication is done through a keyboard and screen), allowing the robot to start the search filtering those objects that fulfill this first description. Based on this subset, ORO provides a descriptor (or set of descriptors) that allows maximum discrimination among objects in the subset. The robot queries the user about the value of the descriptor (or the most discriminant among the set of descriptors) and with this new information, the current subset of objects

human	It is a tableware.
jido	*[retrieve possible objects: coke-bottle, yellow-bottle, orange-bottle, cup-with-handle]*
	[keep visible objects: coke-bottle, yellow-bottle, cup-with-handle]
	[obtain discriminants: type, color.]
	Which type of object it is: bottle or cup?
human	Bottle.
jido	*[obtain possible objects: coke-bottle, yellow-bottle.]*
	[obtain discriminants: color.]
	What color is the object: red or yellow?
human	Red.
jido	*[obtains possible objects: coke-bottle.]*
	The object is the Coke bottle!

Fig. 13 Example of the robot playing Spy game.

is filtered again. The process is repeated until either obtaining a single object that fulfills all the descriptor values, or failing (*i.e.* no object found).

We have integrated the game on the LAAS-CNRS Jido robot. Objects are identified through a tag-based vision approach[18] and motion capture is used for human tracking. Their descriptions regarding categories (type of object) and features (color, shape) are manually given in advance. Spatial relationships (`front`, `back`, `left`, etc, and `in`, `on` and `next to`) and visibility (only visible objects for both agents can be considered in the game) are automatically computed on-line by SPARK. Figure 13 shows an example of a round game.

5.3 *Moving to London* Scenario

In order to illustrate some of different reasoning abilities of the robot presented in this paper, we have designed the following daily life situation. Tom and Jerry are moving to London, so they are packing things in boxes. The scenario takes places in the living room, where Jido is observing while they move things here and there. To assess the reasoning abilities of the robot they ask Jido for information (entered through keyboard). Ideally, the robot should perform actions when required (*e.g.* hand an object when asking "give me..."). However, since it is out of the scope of this work, we do not include any motion from the robot's side. Similar to the *Spy Game* scenario, perception of objects is done through a tag-based system and humans are detected through motion capture. We next describe in detail three situations where we can follow the internal robot's reasoning and the interaction with the users.

[18] ARToolKit: `http://www.hitl.washington.edu/artoolkit/`

5.3.1 Initial Situation Assessment

First Tom enters the room with some of the things they need to pack: a toolbox and two videos. He leaves one of the video tapes ("Jido-E") inside one of the boxes (the cardboard box), and the other one ("Lord of the Robots") on the table. We next describe how the situation assessment takes place at each step, *i.e.* how the ontology is updated with the information obtained from the geometric reasoner SPARK. The initial information in ORO corresponds to:

⟨`table type Table`⟩ ⟨`Tom type Human`⟩
⟨`cardBoardBox type Box`⟩ ⟨`Jerry type Human`⟩
⟨`toolbox type Box`⟩ ⟨`videoTape1 type VideoTape`⟩
⟨`videoTape2 type VideoTape`⟩ ⟨`videoTape2 label "Jido-E"`⟩
⟨`videoTape1 label "Lord of the Robots"`⟩

SPARK detects that there is a cardboard box on the table. It thus sends the fact to ORO:

⟨`cardBoardBox isOn table`⟩

Tom enters carrying several objects (Figure 14a) and places them around. Then he leaves. (Figure 14b). The following facts are computed and sent to ORO:

⟨`toolbox isOn table`⟩ ⟨`toolbox isNextTo videoTape1`⟩
⟨`videoTape1 isOn table`⟩ ⟨`videoTape2 isIn cardBoardBox`⟩
⟨`videoTape1 isNextTo toolbox`⟩

5.3.2 Implicit Disambiguation Through Visual Perspective Taking

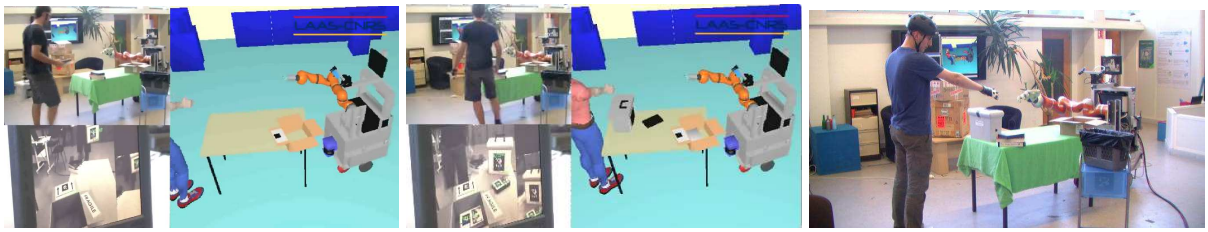
Tom enters the room again while carrying a big box (Figure 1). He approaches the table and asks Jido to handle him the video tape: "Jido, can you give me the video tape". The DIALOGS module queries the ontology to identify the object the human is referring to:

⟨`?obj type VideoTape`⟩

There are two video tapes in the scene: one on the table, and another one inside the cardboard box. Thus, the knowledge base returns both:

⇒ `?obj = [videoTape1, videoTape2]`

However, only one is visible for Tom (the one on the table). Thus, although there is an ambiguity from the robot's perspective (since it can see both video tapes), based on the perspective of its human partner it infers that Tom is referring to the video tape on the table, and not the one inside the box which is not visible from his view. Therefore, non-visible objects are removed obtaining:



(a) (b) (c)

Fig. 14 Situation assessment during initialization (before (a) and after (b) placing the objects on the table). In each image the snapshots correspond to: real environment (top-left sub-image); processed image to identify the tagged objects (bottom-left sub-image); and the 3D environment (right sub-image).(c) Jerry asks Jido for the content of the box while pointing at it.

Robot's beliefs about itself (*robot's model*):
⟨videoTape1 type VideoTape⟩
⟨videoTape1 isOn table⟩
⟨videoTape1 isVisible *true*⟩
⟨videoTape2 type VideoTape⟩
⟨videoTape2 isIn cardBoardBox⟩
⟨videoTape2 isVisible *true*⟩
Robot's beliefs about Tom (*Tom's model*):
⟨videoTape1 type VideoTape⟩
⟨videoTape1 isOn table⟩
⟨videoTape1 isVisible *true*⟩
⟨videoTape2 type VideoTape⟩
⟨videoTape2 isIn cardBoardBox⟩
⟨videoTape2 isVisible *false*⟩

Table 1 Robot's beliefs about itself and its human partner.

```
?obj = [videoTape1]
```

Since only one object is available now, the robot infers that the human refers to it and can eventually execute the command, *i.e.* give it to the human. Alternatively, the robot could first verify with the human whether that was the object being referred to before proceeding to execute the action. Table 1 lists the robot's beliefs about itself and its human partner involved in this situation.

5.3.3 Explicit Disambiguation Through Verbal Interaction and Gestures

In this last situation Jerry enters the living room without knowing where Tom had placed the video tapes. So he first asks Jido: "What's in the box?". Before the robot can answer the question it has to figure out which box Jerry is talking about. Similar to the previous situation, there are two available boxes:

```
⟨?obj type Box⟩
    ⇒ ?obj = [cardBoardBox, toolbox]
```

However both are visible and the cognitive ambiguity resolution cannot be applied. The only option is to ask Jerry which box he is referring to[19]: "Which box, the toolbox or the cardboard box?" Jerry could now simply answer the question. Instead, he decides to point at it while indicating: "That box" (Figure 14c). The robot's perception identifies the `cardBoardBox` as being pointed at and looked at by the human and updates the ontology with this new information using a rule available in the commonsense ontology:

```
⟨Jerry pointsAt carboardBox⟩, ⟨Jerry looksAt carboardBox⟩
                → ⟨Jerry focusesAt carboardBox⟩
```

In the meantime, the DIALOGS module is processing the human verbal input. When trying to resolve the reference "that" it is able to merge[20] both sources of information, verbal and gestural, to distinguish the box Jerry refers to:

```
⟨Jerry focusesAt ?obj⟩
    ⇒ ?obj = [cardBoardBox]
```

Finally, DIALOGS queries the ontology about the content of the box and the question can be answered: "Jido-E". Note that the object's label is used instead of its ID. This way we enhance interaction using familiar names given by the users:

```
⟨?obj isIn cardBoardBox⟩
    ⇒ ?obj = videoTape2
```

At this point Jerry wants to know where the other video tape is, and that is exactly what he asks Jido: "And where is the other video tape?". In this occasion, the DIALOGS module is able to interpret that Jerry is not referring to the video tape which they were just talking about, but to the other one:

[19] Note that Jerry is using the definite article *the*: the robot has to determine which box is relevant. DIALOGS supports other kinds of quantification (existantial – *some* –, universal – *all* – , explicit cardinality), as expressed by the combination of definite/indefinite and singular/plural forms of articles.

[20] Due to synchronization issues, the user should perform the gesture (pointing at) before answering the robot's question and maintain it until the resolution process takes place.

⟨?obj type VideoTape⟩
⟨?obj differentFrom videoTape2⟩
⇒ ?obj = [videoTape1]

Since there is only one possible "other" video tape (there are only two video tapes in the scene), it can directly answer Jerry: "The other video tape is on the table and next to the toolbox."

⟨videoTape1 isOn table⟩, ⟨videoTape1 isNextTo toolbox⟩

6 Conclusion

6.1 Towards an Event-Driven, Knowledge-Oriented Architecture for Personal Robotics

In this paper, we have studied knowledge streams between three components: *(1)* ORO, an ontology-based knowledge server that stores and maintains classified RDF statements produced by other modules in agent-specific models and allows information to be easily retrieved, either through queries or via an event system; *(2)* SPARK, the grounded, human-aware 3D model of the environment that performs all the spatial reasoning within our architecture, including reasoning involving motion planing (to compute reachability of objects) and perspective taking, and *(3)* DIALOGS, a natural language processor that performs simple grammatical parsing of English language, grounds the semantic content of the utterance (if necessary, also interacts with the user to disambiguate), and eventually generates a RDF representation of the sentence.

These components, combined with modules dedicated to symbolic supervision and task planning (these modules are outside of the scope of this article), compose an architecture that we call *knowledge-oriented*:

- Knowledge is explicitly stored in one central and consistent repository of facts, accessible by all modules.
- Knowledge is represented in a strict formalism (OWL statements) and with a clearly defined vocabulary (stated in the `commonsense.oro.owl` ontology).
- The first two points enable both a loosely-coupled architecture where modules can very easily be removed or replaced by other ones as long as they share the same semantics (modules are defined by the knowledge they produce),
- and a *symbolic* reactive, event-driven approach to supervision. By managing events at the same level as the reasoner, we take full advantage of the inference abilities of ORO to trigger events whose `true` conditions can be inferred.
- Finally, this architecture allows for the combination of very different knowledge modalities in a single homogeneous environment, bringing mutual benefits to components. For instance, the dialogue processing module can perfectly run without any geometric perception, but its disambiguation routines can transparently benefit from it when available (since richer symbolic descriptions of objects are then available).

This architecture moves away from standard layered approaches. Interactions between components are mostly bidirectional and we do not have a concept of layers of abstraction (we do, however, have access to the lower level modules of the robot to execute actions, but all cognition-related modules reside at the same level). This is especially visible for the dialogue input processing. This component does not simply act as an alternative perceptual input to the symbolic database, but also actively queries previously acquired knowledge to disambiguate and validate the newly created symbolic knowledge.

Regarding the anchoring question, this architecture is bidirectional. The components we described provide a *bottom-up* grounding process: SPARK and DIALOGS constantly build and push new symbolic contents about the world to ORO where it becomes accessible to decisional layers. In parallel, ORO relies on reasoning in a *top-down* way to produce new facts that may trigger in return physical behaviours.

6.2 Discussion

Our system has however shortcomings and opens several questions on different topics. In this section, we discuss some of these limitations, possible extensions, and how this article contributes to the larger debate on symbol grounding for embodied agent.

6.2.1 Modeling the Real World

The main challenge we address in this work can be formulated as *How to model real-world interaction in a symbolic way, processable by the robot to make decisions*. In the paper we used several times the term *grounding* to describe the process of binding percepts to symbols (later organized in a first-order logic framework). We would like to relate it to Sloman's [37] stance against the *"Symbol Grounding meme"*, where he argues that symbolic grounding is bound to the representation of somatic concepts (*i.e.* roughly, the sensorimotor relationships that the robot learns from its interaction with the world) which in turn severely constraints the domain of concepts accessible to the robot.

We could call this type of grounding *bottom-up* grounding, and Steels [39] claims it is a solved issue.

For us, *grounding* is on the contrary a *top-down* activity: the robot needs to automatically bind a representation (for instance, a word uttered by a human, an image taken from a camera, a sentence extracted from Wikipedia) to an unambiguous, context-dependent, internal concept. This concept may (or may not) be *a priori* available to the robot as a pre-loaded ontology (what we previously called the cultural background of the robot).

Note also that perception issues have been solved in our experiments by using a tag-based object identification method. In section 4.1 we give an example where the human says "the yellow banana is big". It is assumed in the example that the robot already knows about a banana instance that is yellow. In our experiments, this kind of knowledge was either hard coded in scenario-specific ontologies (*e.g.* ⟨`banana_01 type Banana`⟩ where `banana_01` is the id of the banana's tag) or taught to the robot with prescriptive sentences like "Learn that this is a banana" while pointing at the banana's tag. It would be interesting to extend this approach with automatic classifiers (for colour, size, etc.). If the robot later discovers a yellowish and large object, an utterance like "the yellow banana is big" could be used to assert that this object is a banana. A similar approach focused on the combination of visual perception and communication modalities to achieve visual learning has been developed by [43].

While the examples we develop are all based on symbols that have a physical meaning, the system deals equally well with abstract, *exo-somatic*, concepts like *Time*, *Event* or *Place*. Demonstrating this in real experiments would be an interesting development.

Amongst the other shortcomings of our architecture, neither the *domain of validity* nor the context of a fact are represented in a satisfying way (we do store some kind of context –the agent's mental model for instance). This information is meta-information on the knowledge. While the ORO framework allows them through *statement reification*, it does not offer yet a convenient way to store them. One obvious limitation that derives from the lack of efficient meta-knowledge is the absence of knowledge history. With ORO, the robot always lives in the present.

Along the same lines, our current framework lacks a proper management of uncertainty which is essential for real world environments. A probabilistic layer should be added by attaching truth probabilities to statements, similar to [17].

6.2.2 On Thematic Roles and Action Models

The current implementation relies on a small, predefined set of action verbs that can be recognized from natural language (section 4.2). This constraint does not come from the resolution algorithm itself, but rather from the difficulty to automatically extract the thematic roles associated to a verb. This could be improved by linking a symbolic task planner to the DIALOGS module to dynamically provide the list of actions that the robot can process, *i.e.* actions for which the robot can produce a plan. Additionally, we could exploit on-line resources like VERBNET [19], which provides a large machine-processable lexicon of English verbs along with their thematic roles.

6.2.3 Knowledge and Embodiment

The three experiments that were presented in the paper all illustrate how the robot makes use of its embodied nature to establish a meaningful communication with a human. Mainly, because the robot and the human share the same physical environment and they perceive each other, we are able to create a mutual context.

Sloman, in [38], argues however that the strong focus on embodiment in the robotics community has hindered progress towards natural human-robot interaction. Our approach has hopefully made clear that, similar to Beetz et al. [4], we do not consider embodiment *per se* outside of a broader symbolic system, *i.e.* our architecture is not bound to the morphology or the low-level sensori-motor capabilities of a specific agent.

However, we can build a model of the "human point of view" because the robot perceives the human, and is able to estimate, at least partially, what the human perceives or not. We infer that a human focuses on some object because he/she points at it, looks at it, and besides, the object is visible to him. This relies on the embodied nature of the interaction. In turn, this allows us to understand the meaning of sentences like "Give me that".

We hope that this contribution shows that considering embodiment as the most challenging and fruitful characteristic of robotics in regards to the whole AI community does not contradict with a formal, highly symbolic approach of the representation and decision problems that arise in robotics.

Let us conclude this article briefly reviewing and linking Roy's list of challenges for human-robot dialogue with our current approach:

– While more modalities (especially, deictic gestures and social gazes) can be added, we have actually proposed a *cross-modal representation system*.

- One of the main feature of the DIALOGS module is its ability to interactively ground concepts through disambiguation, bringing the ability for the robot to *associate words with perceptual and action categories*.
- The ORO knowledge base offers some support for the *modeling of context*, but a lot remains to be done in this respect.
- *Figuring out the right granularity of models* is partially solved by supporting both a geometric reasoning level and a purely symbolic level. Generally speaking, it appears that complex robotic systems need to operate with a dynamic granularity, depending on the task to achieve.
- *Temporal modeling* is currently missing in our architecture, and symbolic and geometric *planning* is accomplished outside of the knowledge representation loop we presented here. We see planning as an essential tool to build predictive knowledge, and we are looking into this direction.
- Since we provide no time management, our system is currently not able to *match past (learned) experiences with the current interaction*. This ability is obviously a key step for general action recognition, and seems of particular importance for the robot to assess the state of the interaction with the human.
- Finally, Roy mentions *the ability to take into account the human perspective*: this is probably our main contribution which we are now trying to develop even further towards psychology-inspired experiments.

7 Downloads

The various software components presented in this paper are all open-source.

- ORO is the stand-alone ontology server. Download and documentation are on `http://oro.openrobots.org`.
- The *OpenRobots Common Sense Ontology* can be accessed from `http://kb.openrobots.org`.
- SPARK is the 3D-based situation assessment module. Its homepage is `http://spark.openrobots.org`.
- DIALOGS is the natural-language interpretation module. Its homepage is `http://dialogs.openrobots.org`.

Acknowledgements Part of this work has been conducted within the EU CHRIS project (`http://www.chrisfp7.eu/`) funded by the E.C. Division FP7-IST under Contract 215805. We also acknowledge support of Agence Nationale de la Recherche ANR (AMORCES/PSIROB project) and part of this work has been supported by a Marie Curie Intra European Fellowship. We would like to thank as well Patrick Tsemengue and Madhi Chouayakh for their great work on DIALOGS.

References

1. Alami R, Chatila R, Fleury S, Ghallab M, Ingrand F (1998) An architecture for autonomy. The International Journal of Robotics Research 17(4):315
2. Austin J, Urmson J, Sbisà M (1962) How to do things with words. Harvard University Press
3. Baerlocher P, Boulic R (2004) An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. The Visual Computer: International Journal of Computer Graphics 20(6):402–417
4. Beetz M, Mösenlechner L, Tenorth M (2010) CRAM — A Cognitive Robot Abstract Machine for everyday manipulation in human environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems
5. Blisard SN (2005) Modeling spatial referencing language for human-robot interaction. In: in Proc. IEEE Intl. Workshop on Robot and Human Interactive Communication, pp 698–703
6. Brick T, Scheutz M (2007) Incremental natural language processing for HRI. In: Proceedings of the ACM/IEEE international conference on Human-robot interaction
7. Coradeschi S, Saffiotti A (2003) An introduction to the anchoring problem. Robotics and Autonomous Systems 43(2-3):85–96
8. Dale R, Reiter E (1995) Computational interpretations of the gricean maxims in the generation of referring expressions. Cognitive Science 19(2):233–263
9. Daoutis M, Coradeshi S, Loutfi A (2009) Grounding commonsense knowledge in intelligent systems. Journal of Ambient Intelligence and Smart Environments pp 311–321
10. Flavell JH (1992) Perspectives on Perspective Taking, Lawrence Erlbaum Associates, pp 107–139
11. Gruber J (1965) Studies in lexical relations. PhD thesis, Massachusetts Institute of Technology
12. Gunderson J, Gunderson L (2008) Robots, Reasoning, and Reification. Springer
13. Gutiérrez JPM (2001) Directed Motion in English and Spanish, vol 11, Universidad de Sevilla, chap Semantic Role Lists
14. Harnad S (1990) The symbol grounding problem. Phys D 42(1-3):335–346
15. Hawes N, Hanheide M, Hargreaves J, Page B, Zender H, Jensfelt P (2011) Home alone: Autonomous

extension and correction of spatial representations. In: Proceedings of the International Conference on Robotics and Automation
16. Huwel S, Wrede B, Sagerer G (2006) Robust speech understanding for multi-modal human-robot communication. In: The 15th IEEE International Symposium on Robot and Human Interactive Communication
17. Jain D, Mösenlechner L, Beetz M (2009) Equipping robot control programs with first-order probabilistic reasoning capabilities. In: International Conference on Robotics and Automation (ICRA)
18. Kelleher JD, Costello FJ (2009) Applying computational models of spatial prepositions to visually situated dialog. Comput Linguist 35:271–306
19. Kipper K, Korhonen A, Ryant N, Palmer M (2008) A large-scale classification of english verbs. Language Resources and Evaluation 42(1):21–40
20. Kruijff G, Lison P, Benjamin T, Jacobsson H, Zender H, Kruijff-Korbayová I, Hawes N (2010) Situated dialogue processing for human-robot interaction. Cognitive Systems pp 311–364
21. Lemaignan S, Ros R, Mösenlechner L, Alami R, Beetz M (2010) ORO, a knowledge management platform for cognitive architectures in robotics. In: IEEE/RSJ International Conference on Intelligent Robots and Systems
22. Leslie A (2000) Theory of mind as a mechanism of selective attention. The new cognitive neurosciences pp 1235–1247
23. Matuszek C, Fox D, Koscher K (2010) Following directions using statistical machine translation. In: Proc. of Int'l Conf. on Human-Robot Interaction, ACM Press
24. Mavridis N, Roy D (2005) Grounded situation models for robots: Bridging language, perception, and action. In: AAAI-05 Workshop on Modular Construction of Human-Like Intelligence
25. Moll H, Tomasello M (2006) Level 1 perspective-taking at 24 months of age. British Journal of Developmental Psychology 24(3):603–614
26. Nakamura Y (1990) Advanced Robotics: Redundancy and Optimization. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA
27. O'Keefe J (1999) The Spatial Prepositions. MIT Press
28. Regier T, Carlson L (2001) Grounding spatial language in perception: An empirical and computational investgation. Journal of Experimental Psychology
29. Ros R, Lemaignan S, Sisbot EA, Alami R, Steinwender J, Hamann K, Warneken F (2010) Which one? grounding the referent based on efficient human-robot interaction. In: 19th IEEE International Symposium in Robot and Human Interactive Communication
30. Ros R, Sisbot EA, Lemaignan S, Pandey A, Alami R (2010) Robot, tell me what you know about...?: Expressing robot's knowledge through interaction. In: Proceedings of the ICRA 2010 Workshop on Interactive Communication for Autonomous Intelligent Robots (ICAIR), pp 26–29
31. Roy D, Reiter E (2005) Connecting language to the world. Artificial Intelligence
32. Scassellati B (2002) Theory of mind for a humanoid robot. Autonomous Robots 12(1):13–24
33. Searle JR (1976) A classification of illocutionary acts. Language in society 5(01):1–23
34. Searle JR (1980) Minds, brains, and programs. Behavioral and Brain Sciences 3:417–424
35. Shapiro S, Bona J (2009) The GLAIR cognitive architecture. In: AAAI Fall Symposium Series
36. Sisbot E, Ros R, Alami R (2011) Situation assessment for human-robot interaction. In: 20th IEEE International Symposium in Robot and Human Interactive Communication
37. Sloman A (2007) Why symbol-grounding is both impossible and unnecessary, and why theory-tethering is more powerful anyway. Tech. rep., Research Note No. COSY-PR-0705
38. Sloman A (2009) Some requirements for human-like robots: Why the recent over-emphasis on embodiment has held up progress. Creating Brain-like Intelligence pp 248–277
39. Steels L (2007) The symbol grounding problem has been solved. so what's next? Symbols, Embodiment and Meaning Oxford University Press, Oxford, UK
40. Suh I, Lim G, Hwang W, Suh H, Choi J, Park Y (2007) Ontology-based multi-layered robot knowledge framework (omrkf) for robot intelligence. In: IEEE/RSJ International Conference on Intelligent Robots and Systems
41. Tenorth M, Beetz M (2009) KnowRob - knowledge processing for autonomous personal robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems
42. Tversky B, Lee P, Mainwaring S (1999) Why do speakers mix perspectives? Spatial Cognition and Computation 1(4):399–412
43. Vrecko A, Skocaj D, Hawes N, Leonardis A (2009) A computer vision integration model for a multimodal cognitive system. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp 3140–3147
44. Zender H, Kruijff G, Kruijff-Korbayová I (2009) Situated resolution and generation of spatial refer-

ring expressions for robotic assistants. In: Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence, pp 1604–1609
45. Ziemke T (1999) Rethinking grounding. Understanding representation in the cognitive sciences pp 177–190

Biographies

Séverin Lemaignan

Séverin Lemaignan received the M.S. degree in mechanical engineering from the École des Arts et Métiers, Paris, France and the Technical University of Karlsruhe, Germany. After working for a year as research engineer in mobile robotics at INRIA, he is now Ph.D student at LAAS-CNRS, Toulouse, France and Munich Technical University, Germany. He mainly works on knowledge representation for cognitive robotic architectures, natural human-robot interaction and high-level robotic simulation.

Raquel Ros

Raquel Ros is a Marie-Curie fellow at the Robotics and Interactions Group in LAAS-CNRS, where she focuses on implementation of cognitive models for social robots, knowledge grounding and enhancement of human-robot interaction. She received her PhD in computer science from the Universitat Autònoma de Barcelona in 2008. She was then awarded with a Marie-Curie fellowship and joined LAAS-CNRS. Her main research interests are decision-making, cognitive models for social robotics and long-term interaction.

E. Akin Sisbot

Emrah Akin Sisbot received the B.S. degree in computer engineering from Galatasaray University, Istanbul, Turkey, in 2003 and the M.S. degree in artificial intelligence from the University Paul Sabatier, Toulouse, France, in 2004. In 2008 he received his Ph.D degree from University of Toulouse with his research on Human-Aware Motion Planning. Until 2011 he was a Postdoc in Robotics and Interactions Group in LAAS-CNRS where he focused on Spatial Reasoning and Knowledge Management. He recently joined to University of Washington as a Research Associate, where he works on Interactive RGB-D SLAM and robot navigation. His research interests include motion planning, human-robot interaction, spatial reasoning and mapping.

Rachid Alami

Rachid Alami received a Docteur-Ingénieur degree in 1983 in computer science from the Institut National Polytechnique de Toulouse and the “Habilitation à Diriger des Recherches” in 1996 from Paul Sabatier University. He joined the Robotics and Artificial Intelligence group at LAAS-CNRS, Toulouse, France in 1980. He is now a Senior Research Scientist and the head of Robotics and InteractionS group at LAAS-CNRS. His current research interests include the study of control architectures for autonomous robots, task and motion planning, multi-robot cooperation, personal robots, and human-robot interaction.

Michael Beetz

Michael Beetz received his diploma degree in Computer Science with distinction from the University of Kaiserslautern. He received his MSc, MPhil, and PhD degrees from Yale University in 1993, 1994, and 1996 and his Venia Legendi from the University of Bonn in 2000. Michael Beetz was a member of the steering committee of the European network of excellence in AI planning (PLANET) and coordinating the research area “robot planning”. He is associate editor of the AI Journal. His research interests include plan-based control of robotic agents, knowledge processing and representation for robots, integrated robot learning, and cognitive perception.