Artificial Cognition for Social Human-Robot Interaction: An Implementation

Séverin Lemaignan[1,2], **Mathieu Warnier**[1], **E. Akin Sisbot**[1], **Aurélie Clodic**[1], **Rachid Alami**[1]

[1] *LAAS-CNRS, Univ. de Toulouse, CNRS*
7 avenue du Colonel Roche, F-31400 Toulouse, France
firstname.surname@laas.fr

[2] *Centre for Robotics and Neural Systems*
Plymouth University, Plymouth, United Kingdom
firstname.surname@plymouth.ac.uk

Abstract

Human-Robot Interaction challenges Artificial Intelligence in many regards: dynamic, partially unknown environments that were not originally designed for robots; a broad variety of situations with rich semantics to understand and interpret; physical interactions with humans that requires fine, low-latency yet socially acceptable control strategies; natural and multi-modal communication which mandates common-sense knowledge and the representation of possibly divergent mental models. This article is an attempt to characterise these challenges and to exhibit a set of key decisional issues that need to be addressed for a cognitive robot to successfully share space and tasks with a human.

We identify first the needed individual and collaborative cognitive skills: geometric reasoning and situation assessment based on perspective-taking and affordance analysis; acquisition and representation of knowledge models for multiple agents (humans and robots, with their specificities); situated, natural and multi-modal dialogue; human-aware task planning; human-robot joint task achievement. The article discusses each of these abilities, presents working implementations, and shows how they combine in a coherent and original deliberative architecture for human-robot interaction. Supported by experimental results, we eventually show how explicit knowledge management, both symbolic and geometric, proves to be instrumental to richer and more natural human-robot interactions by pushing for pervasive, human-level semantics within the robot's deliberative system.

Keywords: human-robot interaction, cognitive robotics, perspective taking, cognitive architecture, knowledge representation and reasoning

1. The Challenge of Human-Robot Interaction

1.1. The Human-Robot Interaction Context

Human-Robot Interaction (HRI) represents a challenge for Artificial Intelligence (AI). It lays at the crossroad of many subdomains of AI and in effect, it calls for their integration: modelling humans and human cognition; acquiring, representing, manipulating in a tractable way abstract knowledge at the human level; reasoning on this knowledge to make decisions; eventually instantiating those decisions into physical actions both legible to and in coordination with humans. Many AI techniques are mandated, from visual processing to symbolic reasoning, from task planning to *theory of mind* building, from reactive control to action recognition and learning.

We do not claim to address here the issue as a whole. This article attempts however to organise it into a coherent challenge for Artificial Intelligence, and to explain and illustrate some of the paths that we have investigated on our robots, that result in a set of deliberative, knowledge-oriented, software components designed for human-robot interaction.

We focus on a specific class of interactions: human-robot collaborative task achievement [1] supported by multimodal and situated communication. Figure 1 illustrates this context: the human and the robot share a common space and exchange information through multiple modalities (we specifically consider verbal communication, deictic

gestures and social gaze), and the robot is expected to achieve interactive object manipulation, fetch and carry tasks and other similar chores by taking into account, at every stage, the intentions, beliefs, perspectives, skills of its human partner. Namely, the robot must be able to recognise, understand and participate in communication situations, both explicit (e.g. the human addresses verbally the robot) and implicit (e.g. the human points to an object); the robot must be able to take part in joint actions, both pro-actively (by planning and proposing resulting plans to the human) and reactively; the robot must be able to move and act in a safe, efficient and legible way, taking into account social rules like proxemics.

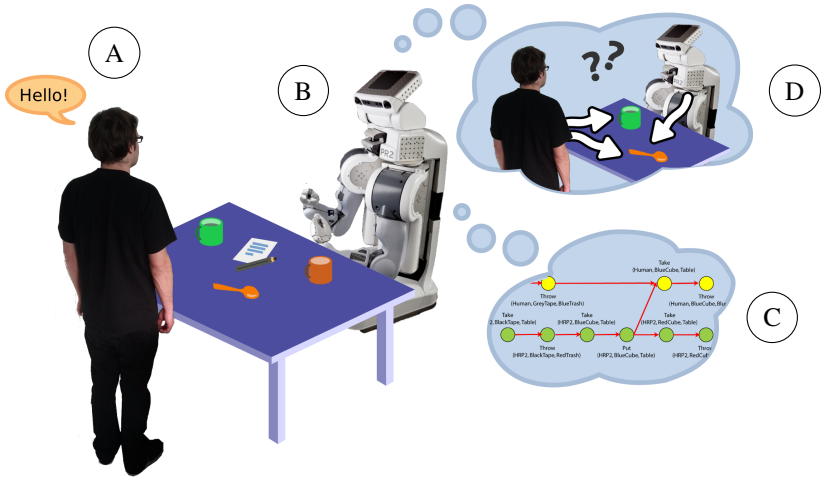


Figure 1: The robot reasons and acts in domestic interaction scenarios. The sources of information are multi-modal dialogue *(A)* and perspective-aware monitoring of the environment and human activity *(B)*. The robot must adapt on-line its behaviours by merging computed plans *(C)* with reactive control. The robot explicitly reasons on the fact that it is (or not) observed by the human. Reasoning and planning take place at symbolic as well as geometric level and take into account agents beliefs, perspectives and capabilities *(D)* as estimated by the robot.

These three challenges, *communication*, *joint action*, *human-aware execution*, structure the research in human-robot interaction. They can be understood in terms of cognitive skills that they mandate. *Joint action*, for instance, builds from:

- a joint *goal*, which has been previously established and agreed upon (typically through dialogue),
- a physical environment, estimated through the robot's exteroceptive sensing capabilities, and augmented by inferences drawn from previous observations,
- a belief state that includes *a priori* common-sense knowledge and mental models of each of the agents involved (the robot and its human partners).

The robot controller (with the help of a task planner) decides what action to execute next [2], and who should perform it, from the robot or the human (or both in case of a collaborative action such as a handover [3, 4]), how it should achieved and what signals should be sensed and/or produced by the robot to facilitate human-robot joint action [5, 6, 7, 8]. It finally controls and monitors its execution. The operation continues until the goal is achieved, is declared unachievable or is abandoned by the human [9].

This translates into several decisional, planning, representation skills that need to be available to the robot [10]. It must be able 1. to represent and manipulate symbolic belief states, 2. to acquire and keep them up-to-date with respect to the state of the world and the task at hand, 3. to build and iteratively refine shared (human-robot) plans, 4. to instantiate and execute the actions it has to perform, and conversely, to monitor those achieved by its human partner.

Besides, such abilities should be designed and implemented in a task-independent manner, and should provide sufficient levels of parametrization, so that they adapt to various environments, different tasks and variable levels of engagement of the robot, ranging from teammate behaviour to assistant or pro-active helper.

These are the challenges that we will discuss in this article.

1.2. Contribution and Article Overview

Our main contributions focus on the architecture of the decisional layer of social robots. Specifically, the deliberative architecture of a robot designed to share space and tasks with humans, and to act and interact in a way that supports the human's own actions and decisions. We present hereafter a model of cognitive integration for service robots that:

- exposes a principled approach to integrate a set of complex cognitive components in an explicit, semantics-oriented and yet loosely-coupled fashion;
- achieves multi-modal and interactive symbol grounding [11] in complex, real-world environments involving one or several humans and a robot;
- distributes the computation of symbolic knowledge by combining perspective taking, affordances computation, situated dialogue and logical inference;
- provides generic mechanisms for the robot to reason about the mental state of its human partners;
- reuses the same set of affordances and inferences, together with explicit contextual reasoning on humans and robot abilities, to generate human-robot shared plans.

This architecture is fully implemented and we demonstrate it on several robotic platforms and in several interaction scenarios. It eventually proves to be an effective framework for novel contributions about human-robot joint action [12, 13, 14], as well as for multi-disciplinary studies [15, 16, 17, 18, 19, 20].

The remaining of the article details this robotic architecture. We organise this discussion in five sections. The next section introduces the architecture as a whole, as well as the knowledge model that we have developed for our robots. Section 3 discusses each of the cognitive components of the architecture. Section 4 presents two studies that illustrate in a practical way what can be currently achieved with our robots. The Sections 5 and 6 finally summarise our main contributions and restate the key challenges that human-robot interaction brings to Artificial Intelligence.

2. Deliberative Architecture and Knowledge Model

2.1. Building a Human-Aware Deliberative Layer

Articulating multiple independent software modules in one coherent robotic architecture is not only a technical challenge, but also a design and architectural challenge. In particular, properly managing the rich semantics of natural interactions with humans raises a range of issues. Our basic assumption and guiding principle is that human-level interaction is easier to achieve if the robot itself relies internally on human-level semantics. We implement this principle by extensively relying on explicit knowledge representation and manipulation: software components communicate with each other using first-order logic statements organised into ontologies and whose semantics are close to the ones manipulated by humans.

Figure 2 gives an overview of our architecture. An active knowledge base (ORO), conveniently thought as a semantic blackboard, connects most of the modules: the geometric reasoning module (SPARK) produces at relatively high frequency symbolic assertions describing the state of the robot environment and its evolution over time. These logical statements are stored in the knowledge base, and queried back, when necessary, by the language processing module (DIALOGS), the symbolic task planner (HATP) and the execution controller (SHARY or PYROBOTS). The output of the language processing module and the activities managed by the robot controller are stored back as symbolic statements as well.

For instance, a book laying on a furniture might be picked up by SPARK and represented in symbolic terms as ⟨`BOOK1 type Book, BOOK1 isOn TABLE`⟩. These symbolic statements are stored in the knowledge base ORO and made available to the other cognitive modules. Later, the robot might process a sentence like "give me another book". The DIALOGS module would then query the knowledge base: `find(?obj type Book, ?obj differentFrom BOOK1)`, and write back assertions like ⟨`HUMAN desires GIVE_ACTION45, GIVE_ACTION45 actsOn BOOK2`⟩ to ORO. This would in turn trigger the execution controller SHARY to prepare to act. It would first call the HATP planner. The planner uses the

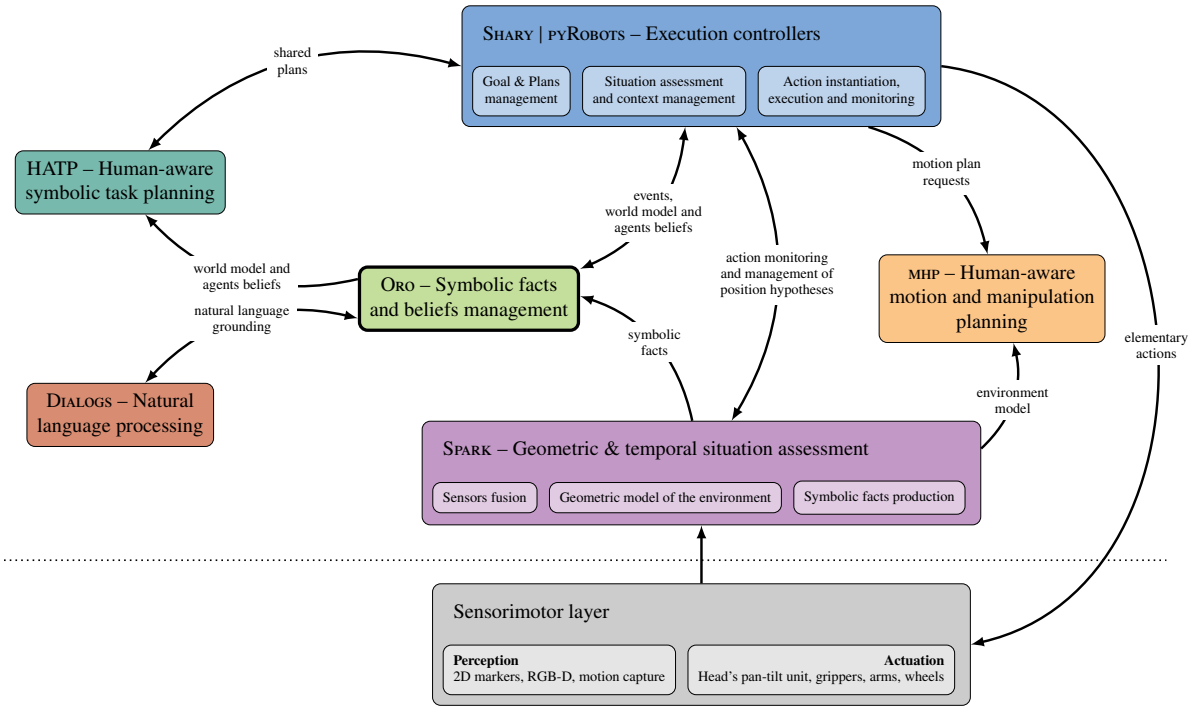


Figure 2: Overview of the architecture. A deliberative layer, composed of six main modules, interacts with a low-level sensori-motor layer. Knowledge is centrally managed in an active *semantic blackboard*, pictured above with a thick border. The links between components depicted on the figure underline the central role of the knowledge base: many of the data streams are actually symbolic statements exchanged through this semantic blackboard.

knowledge base to initialise the planning domain (e.g.`find(BOOK2 isAt ?location)`), and returns a full symbolic plan to the execution controller. Finally, the controller would execute the plan and monitor its achievement, both for itself and for the human. We present complete examples of similar interactions in Section 4.

Our architecture has not been designed to replicate or provide a plausible model of human cognition, and in this sense, we distinguish ourselves from research on *cognitive architectures*. Instead, our main design principle is to nurture the decisional components of the robot with models of human behaviour and human preferences in order to develop an effective artifical cognition for a robot that is able to serve and interact seamlessly with humans. In this sense, it shares its objectives with [21, 22].

At high level, this architecture relies on the same principles as a number of known robotic layered architectures [23, 24, 25, 26]. However, the main point for us was to refine and study in detail the internals of the deliberative layer. In our model, we suggest that the interactions between components within this deliberative level have to be essentially bidirectional. We also suggest not to introduce any sub-layers of abstraction amongst these deliberative components[1]. The natural language processing component illustrates this structure: instead of being an independent input modality whose outputs would be unidirectionally fed to "higher" decisional components, it lives in the deliberative space at the same level as other deliberative components, and make use of the knowledge base in a bidirectional manner, to interpret, disambiguate natural language, and eventually store newly produced interpretations (natural language processing is discussed in Section 3.3). Another example is the intricate relation between high-level symbolic planning and geometric planning to deal with affordances and human preferences.

Our architecture relates to *Beliefs, Desires, Intentions* (BDI) architectures. As put by Woolridge [27], BDI architectures are primarily focused on *practical reasoning*, i.e. the process of deciding, step by step, which action to perform to reach a goal. The management of the interaction between knowledge (the beliefs) and task and plan repre-

[1] We must clarify that we do have lower-level modules to execute actions or manage sensors, but all cognition-related modules live in the same global deliberative space.

sentation and execution (the desires and the intentions) is central, and aims at selecting at each step the best sub-goal. It becomes then an intention that the robot commits to. As for any cognitive system, this fundamental interaction between knowledge and actions is central to our approach as well, and typically involves the dialogue module to acquire *desires* from the other agents, and the planner and the execution controller to first decide to take into account (or not) an incoming desire as a *goal*, and then to generate and manage *intentions* from these goals through the symbolic task planner.

We extend upon BDI architectures by running other background deliberative tasks, without them being explicitly triggered by *desires* in the BDI sense. The main ones include situation assessment, action monitoring and processing of non-imperative speech (including performative dialogue that can possibly change the internal state of the robot, but does not lead directly to the creation of *desires*, like assertion of new facts or question answering).

2.2. Knowledge Model

In our architecture, knowledge manipulation relies on a central server (the ORO server [28], Figure 5 top) which stores knowledge as it is produced by each of the other deliberative components (the clients). It exposes a `json`-based RPC API to query the knowledge base [29]. We represent knowledge as RDF triples in the OWL sub-language [2]. Every time triples are added or removed from the knowledge base, a Description Logics reasoner (PELLET [30]) classifies the whole ontology and inserts all possible inferred triples. The clients of the ORO server are in charge of managing themselves the knowledge (when to add, when to update, when to retract knowledge) as no meta-semantics are carried over that let the server manage itself these dynamics.[3]

This architecture design (a central knowledge base that essentially appears as a passive component to the rest of the system – even though it actually actively processes the knowledge pool in the background to perform inferences) departs from other approaches like the CAST model [31] where knowledge is represented as a diffuse, pervasive resource, or the CRAM/KnowRob architecture [32] where the knowledge base is an active hub that pro-actively queries perceptual components to acquire knowledge. We believe that our design leads to a good *observability* (knowledge flows are explicit and easy to capture since they are centralised) as well as high modularity (modules communicate through an explicit and unified API).

At run-time, the knowledge available to the robot comes from three sources. *A priori* knowledge is stored in an ontology (the OPENROBOTS ontology, discussed hereafter) and is loaded at start-up. This static source implements the *common-sense* knowledge of the robot, and might optionally include scenario-specific knowledge (for instance, about objects that are to be manipulated). The second part of the knowledge is acquired at run-time from perception, interaction and planning. The next sections go into details of these processes. The third source of symbolic statements comes from the inferences produced by the reasoner.

Contrary to similar projects like KnowRob [33] that relies on the concept of *computables* to lazily evaluate/acquire symbolic facts when needed, we have an *explicit* approach where we greedily compute and assert symbolic statements (like spatial relations between objects, see Section 3.2). For instance, whenever a client queries KnowRob to know if ⟨`OBJECT1 isOn OBJECT2`⟩, KnowRob calls a geometric reasoner to evaluate if this specific spatial relation holds at that point in time. With our approach, spatial relations are instead computed and asserted *a priori* by a dedicated process that continuously runs in the background. This design choice trades scalability for explicit reasoning: at any time, the full belief state is made explicit, and therefore provides the reasoner with the largest possible inference domain. This is of special importance for an *event-based* architecture like ours (see Section 3.5), where an explicit and comprehensive belief state is required to not miss events.

2.2.1. RDF as a Formalism for Semantics

The ORO server relies on Description Logics (OWL) to represent and manipulate knowledge. Relying on RDF triples and Description Logics has advantages such as good understanding of its trades-off, thanks to being widespread in the semantic Web community; the availability of mature libraries to manipulate the ontology; interoperability with several major on-line knowledge bases (OPENCYC, WORDNET, DBPEDIA or ROBOEARTH [34] for examples); open-world

[2] `http://www.w3.org/TR/owl2-overview/`

[3] With one exception: so-called *memory profiles* let the server automatically discard facts (i.e. forget about them) after a specific period of time. We present this feature in section 3.1.3.

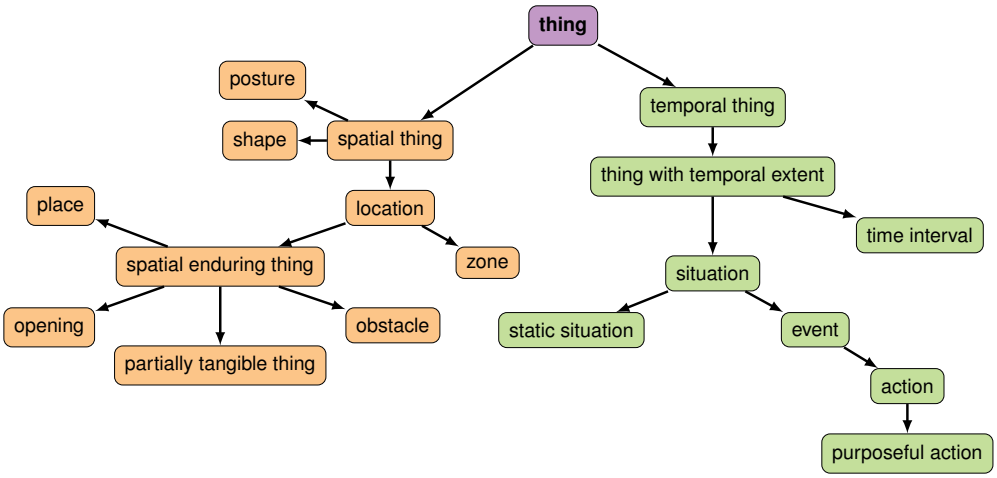


Figure 3: The upper part of the ORO common-sense conceptualization (TBOX). These concepts are shared with the OPENCYC upper-ontology. They relate to each other through 'is-a' subsumption relations.

reasoning (which enables us to represent that some facts might be unknown to the robot); and the formal guarantee of decidability (it is always possible to classify a Description Logics ontology).

It also has restrictions, both basic (the suitability of Description Logics when reasoning on –typically non-monotonic– commonsense knowledge has been questioned) and practical: RDF triples imply binary predicates, which constrains the expressiveness of the system or leads to inconvenient reifications. Alternatives have been proposed (like KNOWROB [33]) that interleave RDF with more expressive logic languages like PROLOG with however other limitations, like closed-world reasoning.

Classification performance is another issue: in our experience, an ontology sized for a typical study (about 100 classes and 200 instances), classification takes around 100ms, which may introduce perceptible delays during interactions. Besides, the performances are difficult to predict: the insertion of seemingly simple statements may change abruptly the logical complexity of the knowledge model and lead to a noticeable degradation of classification time.

This knowledge model also largely excludes representation of continuous phenomena (like time) or uncertain phenomena. When required (for instance for action recognition), these are managed within dedicated components (like SPARK, discussed in Section 3.2), and are not represented in the knowledge base.

Alternative formalisms have been successfully investigated in robotics to address some of these restrictions. Besides the Prolog/OWL combination relied upon by KNOWROB, *Answer Set Programming* has been used in robotics [35, 36] for instance to better supports non-monotonic reasoning. We have also pointed in [37] how modal logics like the epistemic logics could be relevant to the particular field of social human-robot interaction as they allow for the representation of alternative mental models. First-order logic and OWL ontologies have however proved so far a simple, effective and sufficient symbolic framework for our experimental applications.

Incidentally, and because ontologies and RDF statements remain conceptually simple (compared to full logical languages like PROLOG or modal logics), their adoption has also effectively helped to grow awareness amongst colleagues on the significance of the "semantic level" when developing new components for the robot.

2.2.2. The OpenRobots Ontology

As previously mentioned, the logical statements exchanged between the deliberative components are organised within the ORO knowledge base into an ontology. The conceptualization (i.e. the system of concepts or *TBox*) of this ontology is statically asserted[4], while the instantiation (the *ABox*) of the ontology is generally dynamically asserted at run-time, by the other cognitive components.

[4]It can however be altered at runtime, see the "Cat" example in Section 3.3.1.

The *OpenRobots* common-sense ontology represents the statically asserted part of the ontology. It has been designed from two requirements: being practical (i.e. covering our experimental needs) and conforming as much as possible to existing standards (specifically, the OpenCyc upper ontology [38]).

This leads to a bidirectional design process: from *bottom-up* regarding the choices of concepts to model, *top-down* regarding the upper part of the conceptualization. This upper part of the ontology is pictured on Figure 3. All the classes visible in this figure belong to the OpenCyc namespace.

Aligning the upper part of the ontology on OpenCyc (as done by other knowledge representation systems, like KnowRob [33] or PEIS K&R [39]) has multiple advantages. First the design of this part of the ontology is generally difficult: it pertains to abstract concepts whose mutual relations comes to philosophical debates. The upper taxonomy of OpenCyc represents a relative consensus, at least within the semantic Web community. Then, because it is a well established project with numerous links to other on-line databases (like Wikipedia or WordNet), the reuse of key OpenCyc concepts ensures that the knowledge stored by the robot can be shared or extended with well-defined semantics. The concept of *Object* is a good example as it represents a typical case of ambiguous meaning: in everyday conversation, an object is a relatively small physical thing, that can be usually manipulated. A human is not usually considered as an object. OpenCyc however precisely defines an object as anything at least *partially tangible*. This includes obviously the humans, and actually many other entities that would not be commonly said to be objects (the Earth for instance). By relying on well-defined and standard semantics to exchange information between artificial systems, we avoid semantic ambiguities. As the robot interacts with humans, we must however address the discrepancy between OpenCyc concepts and human terminology. In these situations, we manually label the OpenCyc concepts with appropriate human names: for instance, the OpenRobots Ontology associates the label "*object*" to the concept `cyc:Artifact` instead of the concept `cyc:PartiallyTangible`. These labels are used in priority during the grounding of verbal interactions.

Figure 3 also illustrates the fundamental disjunction in the Oro model between *temporal* and *spatial* entities (formally, $(\texttt{TemporalThing} \sqcap \texttt{SpatialThing})^{\mathcal{I}} = \emptyset$, with $\mathcal{I}$ the *interpretation* of our model).

On this same figure, the class `PurposefulAction` represents the superset of all the actions that are purposefully performed by the robot (or another agent). Actual actions (i.e. subclasses of `PurposefulAction` like `Give` or `LookAt`) are not initially asserted in the common-sense ontology. They are instead added at run-time by the execution controller (in link with the symbolic task planner) and the natural language processor based on what the robot is actually able to perform and/or to interpret in the current context (i.e. the current robot configuration and the actions required by the scenario). The set of actions that the robot can interpret usually closely resemble the planning domain in use (i.e. the set of tasks known to the symbolic task planner, with their corresponding pre- and post-conditions).

The tree[5] in Figure 3 is not equally developed in every directions. For example, the subclasses of `PartiallyTangibleThing` (i.e. what we commonly call *objects*) are shown in Figure 4. Our 'bottom-up" design process of the ontology is apparent on this figure: only the subclasses relevant to the context of service robotics in an human-like environment are asserted: For performance reasons as well as clarity, we have decided against an extended conceptual coverage of what "partially tangible things" might be. We instead opportunistically extend the common-sense knowledge when required by studies.

Lastly, the Oro common-sense ontology contains several rules and class expressions that encode non-trivial inferences. The definition of `Bottle` as found in the Oro ontology is a typical example:

$$\texttt{Bottle} \equiv \texttt{Container} \wedge \texttt{GraspableObject} \wedge \texttt{hasShape} \in \texttt{CylinderShape}^{\mathcal{I}} \wedge \texttt{hasMainDimension} \in [0.1, 0.3]$$

If a human informs the robot that a given object is indeed a bottle, the robot can consequently derive more information on the object. Conversely, if the human affirms that "a car is a bottle", the reasoner may detect logical contradictions (like inconsistent sizes) and reject the assertion. The Dialogs module relies on such logical consistency checks when processing natural language inputs, both to ensure that the verbal input has been correctly acquired and parsed, and also to verify that what the human says is logically consistent.

We further discuss the strengths and weaknesses of this knowledge framework in Section 5.4.

[5]While this subset of the ontology is a tree, it does not generally have to be the case. In particular, the concept system (the *TBox*) of the Oro common-sense ontology does not form a tree.

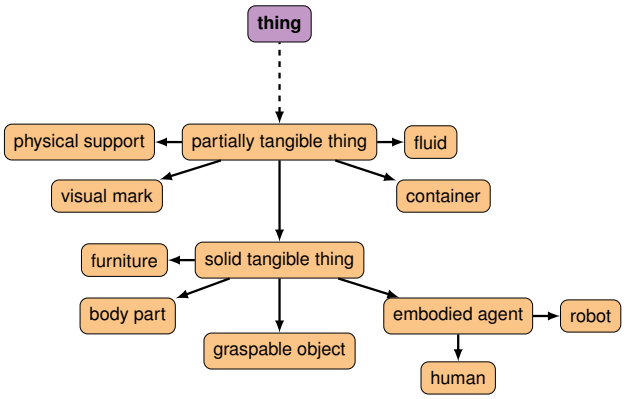


Figure 4: Subclasses of `PartiallyTangibleThing` explicitly stated in the OpenRobots Ontology.

2.3. Symbol Grounding

Grounding (also called *anchoring* when specifically referring to the building of links between *percepts* and *physical objects* [40]) is the task consisting in building and maintaining a bi-directional link between sub-symbolic representations (sensors data, low-level actuation) and symbolic representations that can be manipulated and reasoned about [11]. This represents an important cognitive skill, in particular in the human-robot interaction context: in this situation, the link that the robot has to establish between percepts and symbols must map as well as possible to the human representations in order to effectively support communication.

Symbol grounding connects hence the knowledge model to the perception and actuation capabilities of the robot. The different components that we have mentioned so far exhibit grounding mechanisms: geometric reasoning and dialogue processing modules constantly build and push new symbolic contents about the world to the knowledge base. We detail this process in the next sections.

3. Cognitive Skills

The previous section has introduced the integration model of our architecture, as well as the associated knowledge model. We discuss in this section each of its building blocks. They are pictured in Figure 2, along with their connections to the others components.

We call *cognitive skills* the deliberative behaviours that are 1. *stateful*, i.e. keeping track of previous states is typically needed for the component to perform appropriately; 2. *amodal* in that the skill is not inherently bound to a specific perception or actuation modality; 3. manipulate *explicit semantics*, typically by the mean of symbolic reasoning; 4. operate at the *human-level*, i.e. are legible to the humans, typically by acting at similar levels of abstraction.

We present first the main internal cognitive capabilities, implemented in the Oro knowledge base itself, and then discuss successively the situation assessment module Spark, the dialogue processor Dialogs, the symbolic task planner HATP, and finally the main features of our execution controllers Shary and pyRobots. Note that greater details on each of these modules can be found in their respective publications (the corresponding references are provided hereafter).

3.1. Internal Cognitive Skills

We call *internal* those cognitive capabilities that are tightly bound to the knowledge model, and hence implemented directly within the Oro server. We present here three of them: *reasoning*, *theory of mind* modelling and our (naive) approach to *memory management*.

3.1.1. Symbolic Reasoning

As mentioned in the previous section, we use the Pellet open-source reasoner to reason on the knowledge base. It supports several standard inference mechanisms: consistency checking, concept satisfiability, classification and realisation (computation of the most specific classes that a concept belongs to). In case of logical inconsistency, the reasoner can also provide explanations (we currently only use them for debugging purposes).

Besides, Oro server implements several algorithms (presented in [17]) to identify similarities and differences between concepts (classes or instances): the *Common Ancestors* algorithm, useful to determine the most specific class(es) that include a given set of individuals; the *First Different Ancestors* algorithm that returns what can be intuitively understood as the most generic types that *differentiate* two concepts; and *clarification* and *discrimination* algorithms that play a key role in the process of *interactive grounding* of the semantics of concepts (we discuss this process in section 3.3). Clarification and discrimination algorithms are based on what we call *descriptors*, i.e. properties of individuals, either statically asserted in the common-sense ontology, acquired by the robot through perception or pro-active questioning of the human partner, or derived from other reasoning algorithms like the *Common Ancestors* and *Different Ancestors*. The *discrimination* algorithm consists then in looking for discriminants, i.e. descriptors that allow a maximum discrimination among a set of individuals.

3.1.2. Theory of Mind

Theory of Mind (originally defined in [41]) is the cognitive ability that allows a subject to represent the mental state of another agent, possibly including knowledge that contradicts the subject's own model: for example, a book can be at the same time *visible* for agent A, and *not visible* for agent B. Children develop this skill, which is essential to understand others' perspectives during interactions, around the age of three [42].

From the point of view of interactive robotics, it supposes the robot ability to build, store and retrieve separate models of the beliefs of the humans it interacts with. Our knowledge base implements such a mechanism [28]: when the robot detects that a new human has appeared, it initialises a new independent knowledge model (an ontology) for this human agent. All the ontologies that are created share the same common-sense knowledge, but rely on the estimation of the robot of each agent's perspective for their actual instantiation. For example, the robot can (geometrically) compute that a given book is in its own field of view, but not in the human one (the detail of this computation, called *perspective taking*, is discussed in the next section). The robot updates accordingly the two knowledge models it maintains: the `robot` model is updated with the fact ⟨`BOOK isVisible true`⟩, while the `human` model is updated with ⟨`BOOK isVisible false`⟩. These two logical statements are simultaneously asserted, yet contradict when taken together. Since the two knowledge models are however implemented as two independent ontologies, the contradiction does not actually appear and both the models remain logically consistent.

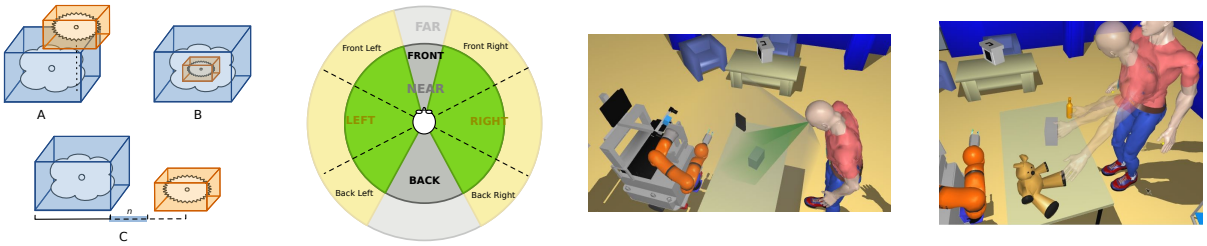
One classical application of this cognitive skill is the so-called *False-Belief* experiment (also known as the *Sally and Anne* experiment, introduced by [43] from an original experimental setting by [44]): a child is asked to watch a scene where two people, A and B, manipulate objects. At some point, A leaves and B hides away one object. When A comes back, we ask the child "where do you think A will look for the object?". Before acquiring a theory of mind, children are not able to separate their own (true) model of the world (where they know that the object was hidden) from the model of A, which contains *false beliefs* on the world (A still thinks the object is at its original position since he did not see B hiding it in a new place). Relying on these separate knowledge models in the knowledge base, we have been able to replicate this experience with our robots [45], in a manner similar to Breazeal et al. [46].

3.1.3. Working Memory

Memory has been studied at length in the cognitive psychology and neuro-psychology communities: the idea of *short-term* and *long-term* memory is due to Atkinson and Shiffrin [47]; Anderson [48] proposes to split memory into *declarative* (explicit) and *procedural* (implicit) memories; Tulving [49] organises the concepts of *procedural*, *semantic* and *episodic* memories into a hierarchy. Short-term memory is eventually refined with the concept of *working memory* by Baddeley [50]. In the field of cognitive architectures, the Soar architecture [51] is one of those that try to reproduce a human-like memory organisation. The GLAIR cognitive architecture [52] also has a concept of long term/short term and episodic/semantic memories.

It is worth emphasising that while memory is commonly associated with the process of forgetting facts after a variable amount of *time*, it actually covers more mechanisms that are relevant to robotics, like priming (concept pre-activation triggered by a specific context [53]) or reinforcement learning.

The Oro server features a mechanism to mimic only minimalistic forms of memory families. When new statements are inserted in the knowledge base, a *memory profile* is attached to them. Three such profiles are predefined: *short term*, *episodic* and *long term*. They are currently attached to different lifetime for the statements (respectively 10 seconds, 5 minutes and no time limit). After this duration, the statements are automatically removed.



a) Agent-independent (allocentric) spatial relations

b) Agent-dependent (egocentric) spatial relations

c) Visibility/Pointing

d) Reachability

Figure 5: Functional overview of the geometric situation assessment module SPARK. SPARK computes symbolic relationships between objects and agents, and exports them to the knowledge base.

This approach is limited. In particular, *episodic* memory should primarily refer to the semantics of the statements (that is expected to be related to an event) and not to a specific lifespan.

We rely however on this mechanism in certain cases: some modules like the natural language processor use the *short term* memory profile to mark concepts that are currently manipulated by the robot as *active concepts*: if a human asks the robot "Give me all red objects", the human, the `Give` action, and every red objects that are found are marked as active concepts by inserting statements such as ⟨`HUMAN type ActiveConcept`⟩ in the short-term memory (which can be considered, in this case, to be a working memory). Likewise, recently seen or updated geometric entities are flagged as `ActiveConcept`. We use this feature during dialogue disambiguation to access concepts recently referred to. On the other hand, our perception layer does not make use of this mechanism. As described in the next section, the environment model of the robot is continuously updated and the derived symbolic knowledge is therefore transient: it lasts only as long as the environment remains in the same state.

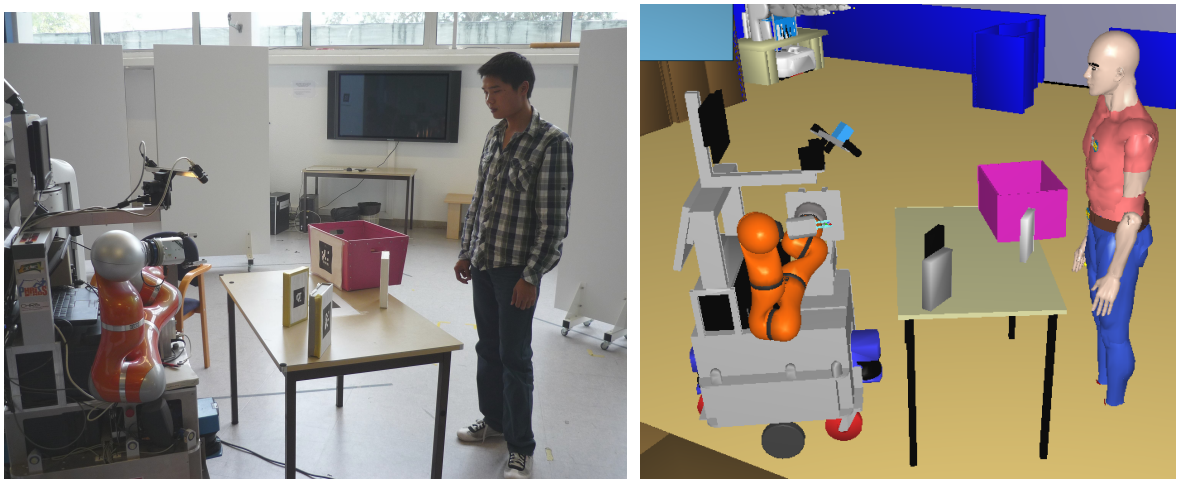
3.2. Acquiring and Anchoring Knowledge in the Physical World

Anchoring perceptions in a symbolic model requires perception abilities and their symbolic interpretation. We call *physical situation assessment* the cognitive skill that a robot exhibits when it assesses the nature and content of its surroundings and monitors its evolution.

Numerous approaches exist, like amodal (in the sense of modality-independent) *proxies* [54], grounded amodal representations [55], semantic maps [56, 57, 58] or affordance-based planning and object classification [59, 60].

We rely on a dedicated geometric and temporal reasoning module called SPARK (*SPAtial Reasoning & Knowledge*, presented in [61]). It acts as a situation assessment reasoner that generates symbolic knowledge from the geometry of the environment with respect to relations between objects, robots and humans (Figures 5 and 6), also taking into account the different perspective that each agent has on the environment. SPARK embeds an *amodal* (as defined by Mavridis and Roy in [55]: the different perceptual modalities are abstracted away into a blended spatial model) geometric model of the environment that serves both as basis for the fusion of the perception modalities and as bridge with the symbolic layer. This geometric model is built from 3D CAD models of the objects, furnitures and robots, and full body, rigged models of humans (Figure 6(b)). It is updated at run-time by the robot's sensors (usually, a combination of vision-based tracking of 2D fiducial markers to identify and localise objects, and Kinect-based skeleton tracking of humans, optionally assisted by motion capture to accurately track the head motion, which is required to compute what the human is looking at).

SPARK runs and continuously updates the knowledge base at about 10Hz. At each step, it re-computes spatial relations and affordances for the whole scene, and send the delta (new relations and relations that do not hold anymore) to the knowledge base. This approach may raise scalability concerns (we did not however observed performance issues in our constrained scenarios, involving typically about 10 objects and two agents) as well as prevent reasoning on the situation history, but simplifies the management of the dynamics of the knowledge (*When do I discard outdated knowledge? When do I update it?*). Since it is equivalent to a reset of the reasoner domain, it also essentially nullifies issues linked to non-monotonic reasoning in the knowledge base (see [62] for a discussion on that question).



(a) Physical setting (b) Corresponding 3D model view

Figure 6: Test setup involving videotapes boxes that are manipulated, with other objects acting as supports or containers. After identification and localisation of the set of objects (using fiducial markers) and acquisition of the position and posture of the human partner (using skeleton tracking), the robot is able to compute that two tapes only are reachable by itself: the black and grey (in the 3D model) tapes. The third tape and the pink container box are only reachable by the human. This physical situation is transformed by SPARK into the set of facts presented in Table 1.

3.2.1. Building an Agent-Aware Symbolic Model of the Environment

Perspective Taking. *Visual* perspective taking refers to the ability for visually perceiving the environment from another person's point of view. This ability allows us to properly handle and represent situations where the visual perception of one person differs from the other one. A typical example found in developmental psychology consists of two similar objects in a room (e.g. two balls) that are both visible for a child, but only one is visible to the adult. When the adult requests the child to hand over "the ball", the child is able to correctly identify which ball the adult is referring to (i.e. the one visible from the adult point of view), without asking [63]. Our architecture endows the robot with such a cognitive skill.

Spatial perspective taking refers to the qualitative spatial location of objects (or agents) with respect to a frame (e.g. *the keys on my left*). Based on this frame of reference, the description of an object varies [64]. Humans mix perspectives frequently during interaction. This is more effective than maintaining a consistent one, either because the (cognitive) cost of switching is lower than remaining with the same perspective, or if the cost is about the same,

Robot's beliefs about itself (*robot's model*)	Robot's beliefs about the human (*human's model*)
⟨PINK_BOX isReachable **false**⟩	⟨PINK_BOX isReachable **true**⟩
⟨WHITE_TAPE isReachable **false**⟩	⟨WHITE_TAPE isReachable **true**⟩
⟨BLACK_TAPE isReachable **true**⟩	⟨BLACK_TAPE isReachable **false**⟩
⟨GREY_TAPE isReachable **true**⟩	⟨GREY_TAPE isReachable **false**⟩
⟨WHITE_TAPE isVisible true⟩	⟨WHITE_TAPE isVisible true⟩
⟨BLACK_TAPE isVisible true⟩	⟨BLACK_TAPE isVisible true⟩
⟨GREY_TAPE isVisible true⟩	⟨GREY_TAPE isVisible true⟩
⟨WHITE_TAPE isOn TABLE⟩	⟨WHITE_TAPE isOn TABLE⟩
⟨BLACK_TAPE isOn TABLE⟩	⟨BLACK_TAPE isOn TABLE⟩
⟨GREY_TAPE isOn TABLE⟩	⟨GREY_TAPE isOn TABLE⟩

Table 1: Symbolic facts computed from the situation depicted in Figure 6. Note how reachability differs for the two agents.

because the spatial situation may be more easily described from one perspective rather than another [65]. Ambiguities arise when one speaker refers to an object within a reference system (or changes the reference system, i.e. switches perspective) without informing her partner about it [66, 67]. For example, the speaker could ask for the "keys on the left". Since no reference system has been given, the listener would not know where exactly to look. However, asking for "the keys on your left" gives enough information to the listener to understand where the speaker is referring to. On the contrary, when using an exact, unambiguous term of reference to describe a location (e.g.. "go north") no ambiguity arises. In SPARK, agent-dependent spatial relations are computed from the frame of reference of each agent.

Symbolic Locations. Humans commonly refer to the positions of objects with symbolic descriptors (like *on*, *next to*...) instead of precise, absolute positions (qualitative spatial reasoning). These type of descriptors have been extensively studied in the context of language grounding [68, 69, 70, 71, 72]. SPARK distinguishes between agent-independent symbolic locations (allocentric spatial relations) and agent-dependent, relative locations (egocentric spatial relations).

SPARK computes three main agent-independent relations based on the bounding box and centre of mass of the objects (Figure 5*a*, [61]): `isOn` holds when an object O_1 is on another object O_2, and is computed by evaluating the centre 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. Note that we do not 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 distance threshold is scaled with the objects' size. Finally, SPARK also computes symbolic facts related to agent independent world dynamics. The predicate `isMoving` states, for each tracked entity, whether it is currently moving or not.

Many other topological relations are dependent from the observation point (egocentric perspective). The predicate `hasRelativePosition` represents the superset of such spatial relations between agents and objects that are agent-dependent. We compute these spatial relations by dividing the space around the referent (an agent) into n regions based on arbitrary angle values relative to the referent orientation (Figure 5*b*). For example, for $n = 4$ we would have the space divided into *front, left, right* and *back*. Additionally, two proximity values, *near* and *far*, are also considered. The number of regions and proximity values can be chosen depending on the context where the interaction takes place. Through perspective taking, SPARK computes for each agent a symbolic description of the relative positioning of objects in the environment.

Table 2 summarises all the symbolic spatial relations computed by SPARK.

3.2.2. Building a Model of Agents

Building a grounded symbolic model of the physical environment does not suffice in general to fully ground the human-robot interaction, and a model of the current capabilities of the agents interacting with the robot is also required.

SPARK computes the following capabilities from the perspectives of each agent:

- *Sees*: this relation describes what the agent can see, i.e. what is within its field of view (FOV). In our current implementation, this affordance is computed by dynamically placing an OpenGL camera at the location of the eyes and running occlusion checks from it. In Figure 5*c* the field of view of a person is illustrated with a grey cone (the wider one). While she is able to see the two small boxes on the table in front of her, the big box on her right is out of her FOV, and therefore, she is not able to see it.

  Besides, SPARK also computes the `seesWithHeadMovement` relation by simulating a small left-right rotation of the head. It represents what an agent *could* see with a minimal effort.

- *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 5*c* shows the field of attention of a person with a green cone (the narrower one). In this example only the grey box satisfies the `looksAt` relation.

- *Points At* holds when an object is pointed at by an agent. This relation is computed by placing a virtual camera on the hand, aligned with the forearm. `pointsAt` is typically used during dialogue grounding, for instance when one of the agents is referring to an object saying "this" or "that" while pointing at it.

We apply an hysteresis filter at the geometric level to ensure a sufficiently stable recognition of these three capabilities.

- *Reachable* enables the robot to estimate the agent's ability to reach for an object, which is instrumental for effective social task planning. For example, if the human asks the robot to give her an object, the robot must compute a transfer point where she is able to get the object afterwards. Reachability is computed for each agent (human or robot) based on Generalised Inverse Kinematics and collision detection. Besides, the robot is able to compute an estimate of the effort needed by an agent to reach an object [73].

Table 2 also lists these abilities, along with the admissible classes for the subjects and objects of the statements.

Subject	**Predicate**	**Object**	*Notes*
`Location`	`isAt` ≡ `cyc:objectFoundInLocation`	`Location`	
	⟶ `isOn` ≡ `cyc:above_Touching`		
	⟶ `isIn`		
	⟶ `isNextTo`		
`Location`	`isAbove` ≡ `cyc:above-Generally`	`Location`	inverse of `isBelow`
			`isOn` ⟹ `isAbove`
`Location`	`isBelow`	`Location`	inverse of `isAbove`
`Location`	`hasRelativePosition`	`Location`	
	⟶ `behind` ≡ `cyc:behind-Generally`		inverse of `inFrontOf`
	⟶ `inFrontOf` ≡ `cyc:inFrontOf-Generally`		inverse of `behind`
	⟶ `leftOf`		inverse of `rightOf`
	⟶ `rightOf`		inverse of `leftOf`
`Object`	`cyc:farFrom`	`Agent`	
`Object`	`cyc:near`	`Agent`	
`Agent`	`looksAt`	`SpatialThing`	
`Agent`	`sees`	`SpatialThing`	
`SpatialThing`	`isInFieldOfView`	`xsd:boolean`	`myself sees * ⇔ * isInFieldOfView true`
`Agent`	`pointsAt` ≡ `cyc:pointingToward`	`SpatialThing`	
`Agent`	`focusesOn`	`SpatialThing`	`looksAt` ∧ `pointsAt` ⇔ `focusesOn`
`Agent`	`seesWithHeadMovement`	`SpatialThing`	
`Agent`	`canReach`	`Object`	
`Object`	`isReachable`	`xsd:boolean`	`myself canReach * ⇔ * isReachable true`

Table 2: List of statements describing agent-independent spatial relationships between objects (top), agent-dependent placements (middle), and attentional states and abilities of agents (bottom). "⟶" indicates sub-properties. Where existing, the equivalent predicate in the OpenCyc standard (prefix `cyc:`) is specified. Note that some relationships are not computed by Spark, but are instead inferred by the reasoner.

3.2.3. Primitive Action Recognition

Monitoring human activity is needed by the execution controllers to track the engagement of the human and the progress of their actions. It is also needed to synchronise seamlessly its own actions with the human actions. Full human action and activity recognition is a task that requires knowledge and reasoning both on high-level facts like goals, intentions and plans, as well as bottom-up data from human and object motions. Spark implements a set of simple temporal and geometric heuristics on human hand trajectories and possible objects placements to recognise simple elementary actions. Those primitive actions are assessed by monitoring situations like "an empty hand is close to an object on a table" (precursor for a *pick*), or "a hand holding an object is over a container" (precursor for a *put*). Spark recognises a set of such primitives. When combined with the other geometric computations and a predictive

Initial human model	*Input*	*Generated query to ontology*	*Statements added to robot model*
⟨BOOK1 type Book⟩	HUMAN1 says:	find(?obj type Book)	⟨HUMAN1 desires SITUATION1⟩
⟨HUMAN1 type Human⟩	"Give me the book"	⇒ ?obj = BOOK1	⟨SITUATION1 type Give⟩
			⟨SITUATION1 performedBy myself⟩
			⟨SITUATION1 actsOnObject BOOK1⟩
			⟨SITUATION1 receivedBy HUMAN1⟩

Figure 7: Processing of a simple, non-ambiguous command, taken from [74]. Thematic roles (performedBy, actsOnObject, receivedBy) are automatically extracted by DIALOGS from the imperative sentence "Give me the book." The resulting statements (right column) are added to the knowledge base, and may eventually trigger an event in the execution controller (see Section 3.5.1).

plan of the human actions (see Section 3.4), the execution controller can track the fulfilment of the pre- and post-conditions of the predicted human actions. The robot relies on these to monitor the engagement of the human and the overall progress of the human-robot shared plan.

3.2.4. Limitations

In its current form, our situation assessment module makes two assumptions: the objects are known in advance (hence, we can rely on proper 3D CAD model for spatial reasoning) and the robot benefits of an nearly perfect perception, made possible by the use of fiducial markers. Each object receives a unique tag which enables an accurate localisation in 3D and prevents recognition ambiguities that would be otherwise reflected in the knowledge base. While SPARK algorithms do not concern themselves with the nature of the input sources, and would work equally well with a full object recognition stack, we did not investigate this research area so far.

Additionally, temporal reasoning (essential for accurate action recognition for instance) is not generally addressed in the current state of our system. Temporal reasoning is used only locally, and does not allow for tracking of long sequences or global events.

3.3. Multi-Modal Communication and Situated Dialogue

3.3.1. Natural Language Grounding

Natural language is a basic interaction modality that we use in our system both as an input (processing of the human speech) and as an output (verbalization of the robot intentions, as well as human-robot shared plans). Natural language processing is facilitated as our architecture manipulates semantics that are close to the human level. This section presents the main features of our speech processor, DIALOGS, that include semantic and multi-modal grounding, and interactive disambiguation. Algorithms and implementation details are provided in [74].

We acquire natural speech input from the human participants through a custom Android-based interface. The interface relies on the Google speech recognition API for speech-to-text (ASR) and relays the textual transcript to the robot. The text is parsed into a grammatical structure (*Part of Speech* tagging) by a custom heuristics-based parser. The resulting atoms are then resolved with the help of the knowledge base to ground concepts like objects (i.e. when a user says "pick up the can", it resolves to which instance of Can the user is referring to) and actions. Figure 7 gives an example of the processing of a simple, non-ambiguous command. The first study (Section 4.1) walks through more complex examples. Heuristics, like the presence of a question mark or the use of imperative mood, are used to classify the sentences into questions, desires or statements. DIALOGS processes these accordingly by answering questions or updating the knowledge base.

The system supports quantification ("give me {a | the | some | all | any | *n*} can"), thematic roles (action-specific predicates that qualify the actions), interactive disambiguation (the robot asks questions when it needs more information), and anaphora resolution ("give *it* to me") based on the dialogue history and the working memory. It also supports knowledge extension by learning new semantic structures. For instance, a sentence like "learn that cats are animals" is converted into ⟨Cat subClassOf Animal⟩ and added to the knowledge base after checking for possible contradictions with existing knowledge. DIALOGS finally interprets common temporal and spatial adverbs (like *above* or *tomorrow*) and translates simple expressions of internal state into *experiences* (for instance, "I'm tired" is processed into ⟨HUMAN1 experiences STATE1, STATE1 hasFeature tired⟩, see also Section 3.5.2). A full account of the DIALOGS features and the corresponding algorithms is available in [74].

3.3.2. Dialogue and Multi-Modality

Because all of the components of our architecture rely on the same RDF formalism to represent their outputs, the different communication modalities are presented in a homogeneous way, as symbolic statements in the knowledge base. This applies both to *explicit* modalities (verbal communication, deictic gestures, gaze focus), and *implicit* modalities (like the body position of the human). The dialogue grounding process makes use of them at two distinct levels to provide multi-modal concept grounding.

First, specific steps of the grounding process explicitly check for the presence and value of certain facts. For instance, when several instances match a category (the human says "give me the bottle" and the robot knows about three bottles), the module may decide to discard some of the candidates based on their *visibility* for the speaker (implicit communication context taking into account the human position). In this particular case, the heuristic is selected by DIALOGS based on the quantifier preceding the class ("give me the bottle"). The first study (Section 4.1) illustrates the details of this process.

As another example, when the human says "this", the robot checks if the human is currently pointing at an object. In that case, *this* is replaced by the object focused on. Otherwise, the robot performs anaphora resolution by looking up in the dialogue history to find a previous concept that the user could refer to.

Note that while the system benefits from complementary modalities, they are not all required. The system can run with the verbal modality alone, at the cost of a simpler interaction. For example, if the human says "this" without the robot tracking what the human points at, no ⟨`HUMAN1 pointsAt ...`⟩ fact is possibly available in the knowledge base, and the robot falls back on the anaphora resolution step alone.

The second level of integration of multi-modality is implicit. By continuously computing symbolic properties from the geometric model, richer symbolic descriptions are available to the system to verbalise or discriminate entities. For instance, the robot may compute that one bottle is next to a glass, while another one stands alone. These symbolic descriptions are transparently re-used in a dialogue context to generate unambiguous references to discriminate between similar objects: "do you mean the bottle that is next to the glass?". The physical context of the interaction is used as an implicit communication modality by DIALOGS. [17] provides a detailed account of our approach to interactive concept clarification and discrimination, along with the related algorithms.

3.4. Human-Aware Task Planning

Whenever necessary, the execution controllers rely on symbolic task planning to convert long-term desires into a set of partially ordered elementary actions. This is the role of the HATP planner (*Human Aware Task Planner*) [75, 76, 77].

The HATP planning framework extends the traditional Hierarchical Task Network (HTN) planning domain representation and semantics by making them more suitable to produce plans which involve humans and robots acting together toward a joint goal. HATP is used by the robot to produce human-robot *shared plans* [78, 79, 80] which are then used to anticipate human action, to suggest a course of action to humans, or possibly to ask help from the human if needed.

The HATP planning domain defines a set of methods describing how to incrementally decompose a task and to allocate subtasks and actions to the robot and/or the human depending on the context. This represents the procedural knowledge of the robot as well as its knowledge about the actions that the human partner is able to achieve. It is stored outside of the central knowledge base, using a specific formalism (see the related discussion at the end of this section).

We discuss next how HATP incrementally builds and synchronises streams of actions for each or the agents (humans and robot) involved in a task, and how it promotes plans that satisfy humans needs and preferences as well as comfort and legibility.

3.4.1. Agents and Action Streams

The originality of HATP resides in its ability to produce *shared plans* which might involve the robot as well as the other participants.

HATP treats agents as "first-class entities" in the domain representation language. It can therefore distinguish between the different agents in the domain as well as between agents and the other entities such as tables and chairs. This enables a post-processing step that splits the final solution (sequence of actions) into two (or more if there are several humans) synchronised solution streams, one for the robot and one for the human, so that the streams may be executed in parallel and synchronised when necessary (Figure 8).

This effectively enriches the interaction capabilities of the robot by providing the system with what is in essence a prediction of the human behaviour. This prediction is used by the robot execution controller to monitor the engagement of the human partner during plan achievement.

The planner also generates causal links and synchronisation points between the robot and his human partners. For instance, in the plan depicted on Figure 8, the human needs to wait for the success of the robot's action `PUTRV`. The robot monitors the success of its own action (by checking for the fulfilment of the action post-conditions; in this particular case ⟨`HUMAN sees BOTTLE`⟩ and ⟨`HUMAN canReach BOTTLE`⟩) to estimate what and when the human is likely to perform his next action (here, `TAKE(BOTTLE, TABLE)`). This, in turn, allows the robot to monitor the human engagement and the progress in the shared plan execution. A complete example is presented in Section 4.1.

The hierarchical structure of the shared plans obtained through the HTN refinement process provide a good basis for plan verbalization. Upon request, it allows the robot to explain to its human partner how a task can be shared [12, 14].

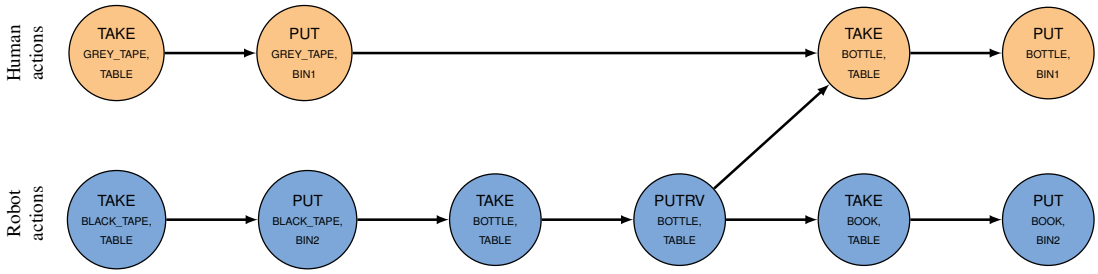


Figure 8: An example of plan produced by HATP for a task consisting in cooperatively moving objects into their associated bins. Two action streams are generated (human actions at the top, robot actions at the bottom, PUTRV stands here for *Put it so it is both Reachable and Visible*). The arrow between the two streams represents a synchronization point between the robot and the human based on a causal link.

3.4.2. Action Costs and Social Rules

A duration and a cost function are associated to each action. The duration function provides a duration interval for the action achievement. It serves both to schedule the different streams and as an additional cost function. Besides time and energy consumption, the cost function integrates factors that measure the satisfaction of the human in terms of acceptability and legibility of the resulting robot behaviour.

HATP includes mechanisms called *social rules* to promote plans that are considered as suitable for human-robot interaction. The following constraints can be set:

Wasted time: avoids plans where the human spends a lot of her time being idle;

Effort balancing with respect of human desires: avoids plans where efforts are not distributed among the agents taking part to the plan, respective of the human preferences. It is indeed sometimes beneficial to balance efforts between the human and the robot. Sometimes the human wants to do more, or on the contrary, prefers to leave the robot do most of the work;

Simplicity: promotes plans that limit as much as possible the interdependencies between the actions of agents involved in the plan, as an issue during the execution of one of those actions would put the entire plan at risk. Also intricate human-robot activity may cause discomfort since the human will find herself repeatedly in a situation where she is waiting for the robot to act;

Undesirable sequences: avoids plans that violate specific user-defined sequences (for instance sequences which can be misinterpreted by the human).

Combining the above criteria, we yield shared plans with desirable interaction features like having the human engaged in a number of tasks while her overall level of efforts remains low, or avoiding having the human to wait for the robot by preventing the action streams from having too many causal links between them.

Figure 9 illustrates such a socially-optimised plan where the *no wasted time* social rule is applied: compared to the plan depicted in Figure 8, the robot first moves the bottle so that the human can immediately take it and put it into the bin, thus reducing the human idle time.

In the current implementation, the social rules are effectively implemented as filters applied to the set of all possible plans computed by the planner. In the future, we intend to extend the plan-search algorithm and integrate the social rules in the process itself.

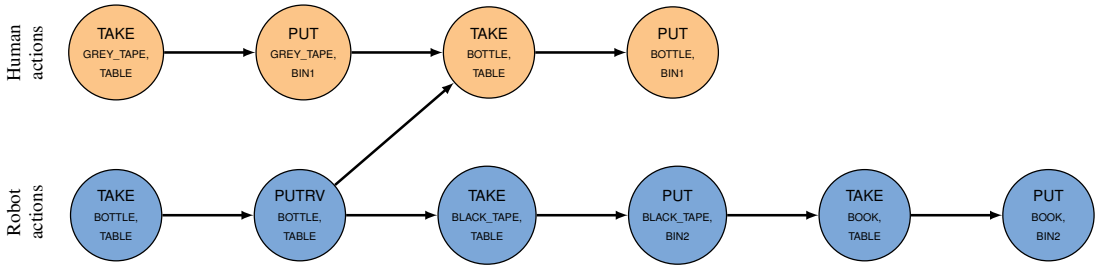


Figure 9: An alternative plan for the task presented in Figure 8 where the *no wasted time* social rule is used to optimise the total duration of the task.

As HATP is a generic symbolic task planner and does not enforce any abstraction level for the planning domain, we have designed a planning domain made of top-level tasks whose semantics are close to the one used in the human-robot dialogue: the planner domain effectively contains concepts like `Give`, `table`, `isOn`. This leads to an effective mapping between the knowledge extracted from the situation assessment or the dialogue, and the planner.

Due to expressiveness issues, we do not represent the planning domain (i.e., the set of tasks with their pre- and post-conditions) in the knowledge base directly (see appendix B of [81] for the full rationale). This effectively leads to independent declarative (the Oro knowledge base) and procedural (the planner) knowledge stores.

3.5. Robot Execution Control

While parts of the architecture (Spark, Oro) have been deployed with external execution controllers (like Cram [32] or the BERT platform [82], as reported in [28]), we have also developed dedicated robot controllers which integrate into the deliberative architecture introduced in Figure 2. Shary [83] is the main one, written in the *Procedural Reasoning System (PRS)* language [84]. We have also developed the Python-based pyRobots [85] that provides a large set of high-level actions and an event-based architecture well suited for prototyping. They both rely on extensive integration with the knowledge base, that serves as the primary source of semantics for the decision making process.

One of the main roles of Shary is to control the production and execution of shared plans. This means essentially context-based refinement and execution of actions, as well as monitoring of those achieved by its human partner. One of the key design goals is to build such abilities in a generic way, and to provide several levels of parametrisation allowing to adapt to various environments and various levels of involvement of the robot, ranging from teammate behaviour to assistant or proactive helper. Based on this, the robot controller invokes the adequate human-aware planners and react to events triggered by the Oro knowledge base (as described below).

Execution control systems for social robots are expected to account not only for the task achievement but also for the communication and monitoring acts needed to support human-robot interactive task achievement [86, 87, 5]. Shary supports as well such a mechanism [88]. It allows to bind action execution to *communication policies* in order to produce multi-modal signals towards the human and to react to human actions and signals. The *communication act* is the central concept in this formalism. It represents an information exchange between the two agents and plays the role of a transition condition. This exchange can be realised through speech, through an expressive motion or a combination of the two. It enables the robot as well as the human to communicate their beliefs about the task to be achieved, in order to share mutual knowledge and to synchronise their activities. This is done through real-time task-based situation assessment achieved by the combination of Shary monitoring processes and Oro inference mechanisms. In Shary, each action description contains not only how to execute it and monitor it, but also how to initiate it (e.g. in the case of a shared action, how both agents need to commit to the task) and how to suspend (or cancel) it. In each of these cases, the action description also makes explicit how the transition has to be communicated to the human partner if necessary. This is supported by a description of the beliefs of each of the agents regarding the action achievement process, as well as a set of so-called *monitoring processes* that update them.

Two recent developments have further improved the robot's ability to adapt to human action, both at the action level and at the plan level. In [89], an extension is proposed for the system to estimate the intentions of the human during collaborative tasks using POMDP models. The controller adjusts then appropriately the chosen set of actions. In [13], the controller is complemented with a framework that allows the robot to estimate the mental state of the human partner, not only related to the environment but also related to the state of goals, plans and actions. They are then accounted for by the execution controller.

3.5.1. Event-Driven Control

The ORO server supports two paradigms to access its content: RPC-style queries (based on the standard SPARQL language) or events. A module can subscribe to an event by registering an event pattern (in its simplest form, a partial statement like ⟨`? type Book`⟩) and a callback. Each time a new instance of a book appears in the knowledge base, the callback is triggered.

This allows us to write reactive robot controllers with a high level of expressiveness: for instance, by subscribing to the event ⟨`HUMAN1 desires ?action, ?action type Give, ?action actsOnObject ?obj, ?obj type Book`⟩, we effectively trigger a behaviour when the human expresses (through dialogue, gestures...) that he wants the robot to give her a book.

The robot controller designer does not need to directly care about how this *desire* is produced (this is delegated to perception modules), he can instead focus on the semantics of the desire.

Note that we transparently take advantage of the reasoning capabilities of the system to trigger event to trigger events: for example, the type of the object (⟨`?obj type Book`⟩) may not be explicitly asserted, but inferred by the reasoner based on other assertions.

3.5.2. Complying with Human Desires and Experiences

We divide the interaction situations perceived from the situation assessment and the communication components into two categories: *desires* (related to *performative acts* in Austin's classification of speech acts [90]) and *experiences*.

Desires are typically human commands ("Give me that book"). The nature of the desired action (to pick, to give, to look, to bring, to show...), along with the action parametrization (thematic roles) are fetched by the execution controller in the knowledge base, and are either sent as goals to the task planner, or executed if the elementary action is directly available.

Experiences, on the other hand, comprise of emotions, states and questions (when asking a question, we consider the human to be in an *interrogative state*). When the knowledge base states that an agent *experiences* a particular emotion or state, the execution controller may decide to handle it, typically by trying to answer the question or using the emotional or physical state as a parameter for subsequent actions. As an example, when the speaker says "I feel tired", we change the motion planner parametrization to lower the effort the human needs to provide for the following joint manipulation tasks.[6]

3.5.3. Human-Aware Action Refinement

Before being executed, elementary actions are further refined by the execution controller with the help of a set of dedicated human-aware geometric motion planning functions provided by a component called MHP (see architecture in Figure 2).

These functions are designed to plan navigation [91] and manipulation [92, 93] paths not only safe but also comfortable and socially acceptable by reasoning explicitly on human's kinematics, vision field, posture and preferences [94, 95, 96]. They also provide routines to compute spatial placements for robot and objects that obey constraints related to the interaction, like optimal mutual reachability or optimal visibility [97].

Finally, when an action requires the motion of both the human and the robot, MHP can plan for both of them in order for the robot to take the lead by automatically computing where the interaction might preferably take place [3, 4]. This can effectively smoothen the interaction by off-loading a part of the cognitive load of the interaction from the human to the robot. As such, this is the geometric counterpart of the HATP symbolic task planner: it is able to make use of a set of social rules to adapt geometric plans to social interactions.

[6] Note that this specific example has been implemented as a proof-of-concept. A broader framework that would support action alteration based on the user's experienced states remains to be investigated.

Study	Focus	Reference
Point & Learn (2010)	Interactive knowledge acquisition	[28]
Spy Game (2010)	Interactive object discrimination	[17]
Interactive Grounding I (2011)	Multi-modal interaction, perspective taking	[98]
Roboscopie (2011)	Human-Robot theatre performance	[99]
Cleaning the table (2011)	Complete architecture integration	[100]
I'm in your shoes (2012)	False beliefs	[45]
Give me this (2012)	Natural joint object manipulation	[101]
Interactive Grounding II (2012)	Multi-modal interaction, perspective taking	[102]

Table 3: Main studies conducted with our cognitive architecture.

4. Support Studies

Our architecture has been deployed and tested in a number of studies on several robotic platforms. Table 3 lists the most significant ones, with their main focuses and reference publications.

We present here elements of two of them in order to illustrate in a practical way the different aspects of the architecture. The first one is focused on knowledge representation and verbal interaction: the human asks for help to find and pack objects (*Interactive Grounding I* in Table 3). The second one (*Cleaning the table*) involves the SHARY execution controller and the HATP symbolic task planner. In this scenario, the human and the robot need to cooperatively remove objects from a table. Robot behaviours and motions are fully planned and then executed.

4.1. Interactive Grounding

This first study is based on a "home move" backstory: two users are moving their belongings to a different home, and need the help of a robot to pack. *Jido*, a single-arm mobile manipulator, is observing while they carry over boxes (Figure 10), and answers questions concerning the location of specific objects. This study focuses on multi-modal, interactive grounding only: the robot observes, builds and maintains knowledge about its human partners perspectives and affordances but does not actually perform any physical action besides verbal interaction and simple head movements.

Objects are perceived through 2D fiducial markers attached to them, and humans are tracked through motion capture. The robot knowledge base is initialised with the ORO commonsense ontology. We next describe two situations where we can follow the internal robot's reasoning and the interaction with the user.

(a) Interactive grounding in a cluttered environment. (b) Disambiguation through pointing.

Figure 10: A scenario involving multi-modal, interactive grounding: the humans can refer to invisible or ambiguous objects that the robots anchor to physical objects through multi-modal interactions with the user.

Implicit disambiguation through visual perspective taking. User A enters the room while carrying a large box (Figure 10(a)). He approaches the table and asks Jido to hand him over a video tape: "Jido, can you give me the video

tape". The DIALOGS module processes this sentence, and queries the ontology to identify the object the human is referring to: `find(?obj type VideoTape)`.

Two video tapes are visible to the robot: one on the table, and another one inside the cardboard box. The knowledge base returns both: `?obj = [BLACK_TAPE, WHITE_TAPE]`.

However, only one is visible to User A (the one on the table). Although there is an ambiguity from the robot's perspective, the human referred to the video tape using the definite quantifier *the*: this is interpreted by the natural language processor as the human referring to a known object, i.e. the one visible in the human's knowledge model.[7]

Explicit disambiguation through verbal interaction and gestures. In the second situation, User B enters the room without knowing where User A had moved the video tapes (Figure 10(b)). He first asks Jido: "What's in the box?". The robot first needs to ground the word "box". Similar to the previous situation, two boxes are visible: `find(?obj type Box)` $\Rightarrow$ `?obj = [CARDBOARD_BOX, TOOLBOX]`

However both are visible to the human and the previous ambiguity resolution procedure can not be applied. The robot generates a question (using the *Discrimination* algorithm, Section 3.1.1) and asks User B which box he is referring to by verbalizing the following question: "Which box, the toolbox or the cardboard box?" User B could answer the question, but he instead decides to point at it: "This box" (Figure 10(b)). SPARK identifies the `CARDBOARD_BOX` as being pointed at, as well as looked at, by the human and updates the ontology with this new information. The reasoner applies a rule available in the common-sense ontology $\langle$`pointsAt(?ag, ?obj)` $\wedge$ `looksAt(?ag, ?obj)` $\rightarrow$ `focusesOn(?ag, ?obj)`$\rangle$. The DIALOGS module merges both sources of information, verbal ("this") and deictic, by issuing a query to the knowledge base that eventually lifts the ambiguity:

```
find(?obj type Box; USER_B focusesOn ?obj)
    ⇒ ?obj = [CARDBOARD_BOX]
```

Finally, DIALOGS queries the ontology about the content of the box and the question can be answered: "Wall-E" (the label of the object had been statically asserted in the ontology at start-up).

At this point User B wants to know where is the other tape: "And where is the other tape?". DIALOGS is able to convert "the other tape" into a new query using the `differentFrom` OWL predicate:

```
⟨?obj type VideoTape⟩
⟨?obj differentFrom WHITE_TAPE⟩
     ⇒ ?obj = [BLACK_TAPE]
```

Since there is only one possible "other" videotape, no specific disambiguation is required. The referent is uniquely identified and DIALOGS can then query for its location: `find(BLACK_TAPE isAt ?loc)`. The robot finally verbalises the result ⟨`BLACK_TAPE isOn table, BLACK_TAPE isNextTo TOOLBOX`⟩ into "The other tape is on the table and next to the toolbox."

4.2. Collaborative Task Planning

This second study demonstrates a richer decision-making process where the ORO server is used in conjunction with the HATP symbolic task planner and the SHARY execution controller to produce and execute a shared plan. The task consists in cooperatively cleaning a table by moving objects into their target bins (Figure 11).

Figure 12 walks through a simplified version of the whole task. It depicts a run with a single video tape on a table. The video tape is reachable by the robot only, while the bin (where the objects are supposed to be eventually moved to) is reachable by the human only: the robot needs to come up with a shared plan that involves a joint action.

The goal is first received by the execution controller (after processing of the user request by the DIALOGS module, not shown on the figure). At t_1 on Figure 12, the video tape is computed by the robot as being reachable by the robot only (columns *Perception* and *Knowledge*), and the execution controller invokes the task planner, which produces a joint plan (column *Plan*) to move the tape so that the human can pick it and drop it into the bin.

[7] Other heuristics are available to the DIALOGS module: for instance, if a tape had been recently mentioned in the dialogue, this instance would have been selected instead as the referent.

Figure 11: The face-to-face setup of the *Clean the Table* study. The physical situation, the SPARK model, and the current step of the plan are visible on the picture.

The first task (`TAKE(GREY_TAPE, TABLE)`) is instantiated by checking that the task pre-conditions hold (in particular, ⟨`GREY_TAPE isOn TABLE`⟩ must be true), and calling the 3D motion planner (column *Actions*, left). The motion planner returns two elementary actions (`PICK_GOTO` followed by `TAKE_TO_FREE`) that the controller executes (by first reaching for the object, grasping it and bringing it back to a free position). The robot's perception monitors the evolution of the scene until the task's post-conditions are verified (at t_2, by satisfying the statement ⟨`ROBOT hasInHand TAPE`⟩), and the next task is then started (placing the tape so that it is reachable to the human).

At t_3, the video tape is now reachable to the human, and the next tasks (taking the tape and placing it in the bin) have to be performed by the human: the robot instructs the user to do so (verbal interaction not represented on the figure) and monitors the actions of the human to detect when the tasks' post-conditions are satisfied (column *Actions*, right). When these post-conditions are fulfilled, the goal is considered to be achieved.

This example illustrates how the symbolic facts are produced by the situation assessment module SPARK, and, in parallel, used by the execution controller to assess the overall progress of the plan.

5. Discussion: When Artificial Intelligence Enables Human-Robot Interaction

The previous sections provide a perspective on a complete deliberative architecture for social robots, including its implementation, supported by experimental deployments. This section synthesises what we see as the main challenges that human-robot interaction brings to Artificial Intelligence. We first discuss how embodied cognition is an essential challenge in human-robot interaction; we rephrase then the requirements of joint actions in terms of five questions; we discuss the importance of building and maintaining a multi-level model of the human; and we finally reflect on the importance of explicit knowledge management in robotic architectures that deal with human-level semantics and state in that respect the current limits of our logic framework.

5.1. Embodied Cognition

Robotics is traditionally regarded as the prototypical instance of *embodied* artificial intelligence, and this dimension is especially prevalent in human-robot interaction, where the robot has to share and collaborate in a joint physical environment. This leads to a tight coupling between the symbolic and the geometric realms: while AI at its origins was mostly concerned with symbolic models, it has been since recognised that not only the mind is not a purely abstract system, disconnected from the physical world, but even more, cognition fundamentally relies on its relation to the physical world (so-called *embodied cognition*). Varela [103] is one of the main discoverer of these mechanisms, and coined the concept of *enactivism* as the theoretical framework that studies the links between cognition, embodiment and actions. This has since been thoroughly studied in robotics and artificial intelligence (Pfeifer and Bongard [104] is one of the reference).

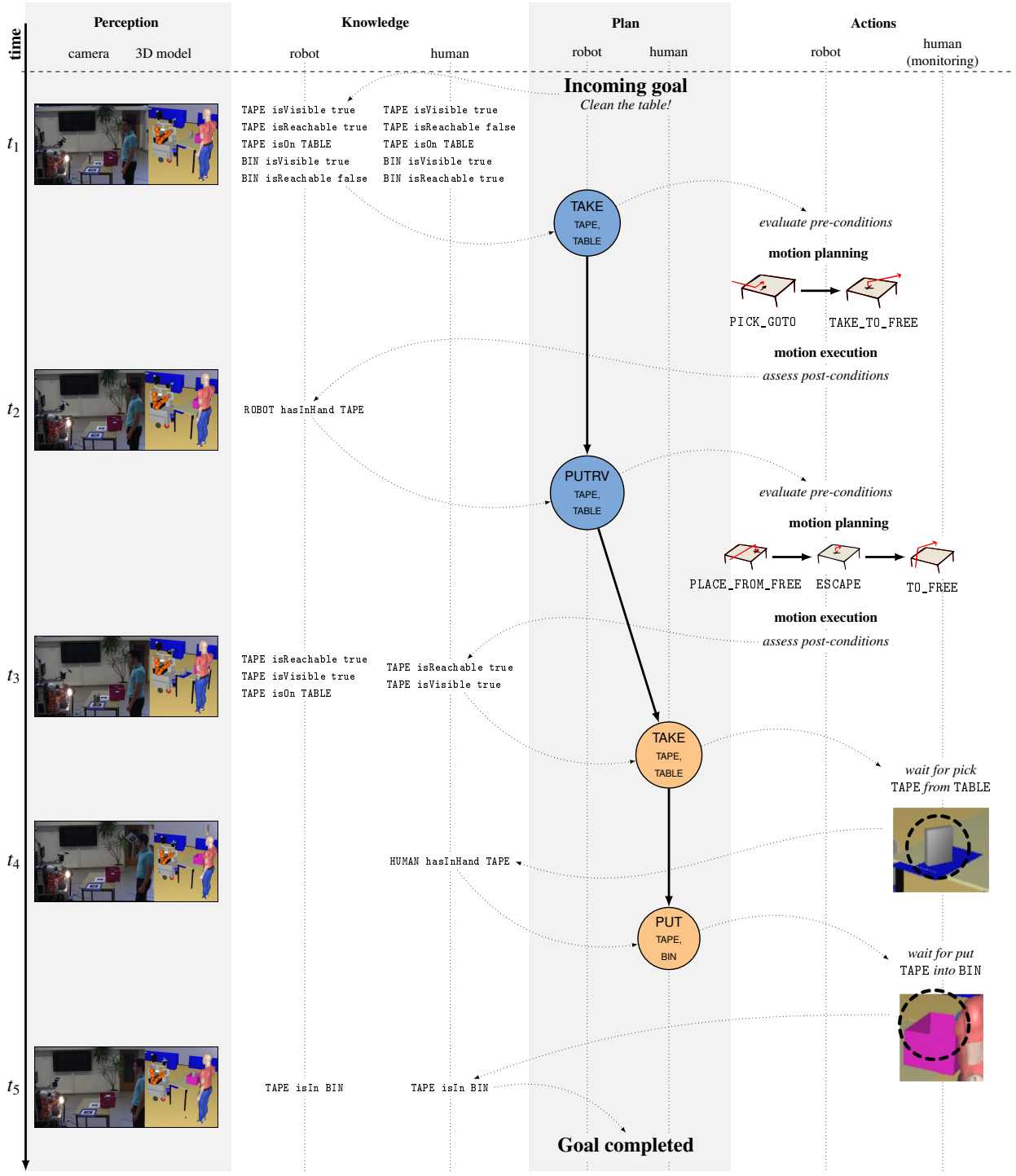


Figure 12: Timeline of the *Cleaning the table* study, presented here in a simplified version: a single object, `TAPE`, must be removed from the table and dropped in the bin `BIN`. The light arrows sketch the global execution flow.

The challenge of symbol grounding is also tightly linked to this issue. It corresponds to the identification or creation, and then, maintenance of a link between the symbol (the syntactic form of knowledge that the computer manipulates) and its semantics (its meaning, anchored in the world). The relation between the symbol, the referent of the symbol, and mediating minds is classically referred as the *semantic triangle* and has been extensively studied in linguistics. The issue of grounding is well known in cognitive science and is summarised by Harnad [11] by this question: "how the semantic interpretation of a formal symbol system can be made intrinsic to the system?". This issue has a major practical importance in robotics: for a robot to be both endowed with a symbolic representational and reasoning system, and able to *act* in the physical world, it must ground its knowledge.

As we have seen, grounding is implemented at different levels in our architecture. The main source of grounded symbolic knowledge is the situation assessment module, SPARK. It builds symbolic facts from spatial reasoning, and also relies on (limited) temporal reasoning to track the world state and build explanations to interpret unexpected perceptions (like an object that suddenly disappears). Because SPARK also tracks humans, which enables perspective-aware spatial reasoning, it can produce as well grounded symbolic knowledge for the agents it interacts with. This is a typical *embodied* cognitive skill.

Grounding also occurs during verbal and non-verbal communication. The DIALOGS module grounds new concepts introduced by the speaker by asking questions until it can attach the new concept to concepts already present in the knowledge pool (if one asks the robot "bring me a margarita", the robot may initially asks "what is a margarita?". The user would answer "a cocktail" and the robot would continue the grounding – "what is a cocktail?" – until it anchors the new concept to ones it already knows like `Drink`). Embodied interactions (like gestures) are also taken into account at this level: we have presented how pointing, for instance, is used by the robot to ground *"this"* or *"that"* to the pointed artifact.

Note that, because only objects marked with 2D fiducial markers are currently recognised (typically, about ten of them are simultaneously used in a given experiment), our grounding mechanisms have only been exercised in small-sized closed world. This simplifies the task, and we can not claim that our approach provides a generic grounding capability. Context, cultural background, "naive physics" knowledge, emotional state of the human are some of the numerous determinants beyond the perception of geometric entities. They would need to be accounted for when grounding human-robot interaction.

5.2. The W-questions of Joint Action

Our context, where a robot has to achieve a task together with humans, raises another set of specific issues to be tackled by the robot's decisional components. We summarise them as the "W-questions": *What*, *Who*, *When*, *Where* and *How*?

What to do next, at different levels of abstractions and while taking into account not only the current state but also the long term goals, is the basic question for an intelligent robot. It is made more complex here by having to deal with the partially observable physical and mental state of the human partner, and by the extended set of possible actions.

Who should act now is of key importance and also needs to be decided upon by the robot. It is sometimes expected as an intelligent behaviour for the robot to wait and let its human partner act instead. Correct management of *turn-taking* leads to various decisional challenges.

When to perform a given action is made more complex by the presence of humans. The robot has to take into account her needs, her rhythms and pace, and her mental state. While performing its share of the task, the robot has to produce signals directed towards the human and to respond to signals produced back at the proper pace to ensure collaboration.

Where to perform an action plays an important role as well: the choice is not trivial and might need elaborated decision. The robot is expected to take into account effort sharing, visibility of its action by the human, disturbance or discomfort induced by its action.

How to perform an action, finally, needs to be reflected upon by the robot: several options to perform an action or to achieve a goal are often available, and selecting one is a non-trivial decision problem. Cost-based planners augmented with social rules are one possible approach: they search for plans that satisfy an acceptable cost in terms of *acceptability* or *legibility* as well.

These five questions should not be considered independently from each other and often require, on the contrary, to be dealt with in a single decision step. The human-aware task and motion planners which we have built are instances of systems which have been designed to deal with such intricate decision issues.

They are in fact considered in each of the components, at different levels of abstraction (from the abstract shared plan level to the action refinement and execution level). At every interaction step, the choice of the next action, of who has to act (the human, the robot or both of them) and when to perform it is made by SHARY with the help of HATP, based on perspective-taking and situation assessment (SPARK and ORO), as well as on the estimation of the human mental state (SHARY with the help of ORO). The actual realization of the action is eventually supported by a full set of on-line planning functions provided by MHP that computes not only trajectories but also the pertinent placements and postures based not only on geometric information but also estimation of the human current mental state and preferences (ORO).

5.3. Putting the Humans into Equations

The correlate of these five *W-questions* is the issue the *human models*: taking appropriate decisions with and in presence of humans requires appropriate models of the human: what the human *can do*, *would like to do*, *knows*, *could infer*, etc.

While the task of describing all the (dynamic) human models that are useful to robots is immense (if doable at all), we claim that it is possible to devise and use such models in limited, but still interesting and useful, contexts such as collaborative human-robot objects manipulation, fetch-and-carry and associated activities in home or work environments.

In our architecture, perspective taking, for instance, is tightly connected to the symbolic knowledge models, and since our knowledge base allows for storage of one knowledge model per agent, we have been able to endow the robot with a simple theory of mind (as explained in section 3.1.2): we explicitly model what the robot knows about its partners in a symbolic way. This knowledge is then re-used in different places, to correctly interpret what the human says, or to plan tasks that are actually doable for the human.

The cognitive model that the robot builds for the agents it interacts with remains today simple and mostly focused on geometric features and affordances (*who sees what? what are our relative positions? what is reachable to whom?*). Extending this knowledge with more subtle perceptions (emotional state for instance) remains to be explored beyond simple examples like the processing of explicit verbal statements like "I'm tired!" (Section 3.5.2).

Motion and action execution also requires human models, and the one we use embeds human preferences and physical constraints that need to be accounted for when synthesising robot motion or producing robot plans. This includes proxemics (human-robot distance) and associated issues (visibility) but also legibility and acceptability criteria expressed in terms of *social rules* that the produced plans should satisfy.

5.4. Explicit Knowledge for Social Robotics

As thoroughly presented in this article, we have built the decisional capabilities of our robots around this idea of explicit knowledge manipulation.

Explicit knowledge in our architecture. The components that we have presented so far build a *knowledge-oriented* architecture: knowledge is explicitly stored in one central and consistent repository of facts, accessible to all modules. It relies on a strict formalism (OWL statements), with a well defined vocabulary (stated in the common-sense ontology). These first two points lead to a loosely-coupled architecture where modules can be removed or replaced by other ones as long as they share the same semantics: modules are defined by the knowledge they produce or consume.

Also, we adopt a symbolic, hybrid (reactive and planning-based), event-driven approach to robot control. 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 (possibly indirectly) inferred using human-level semantics.

And finally, this architecture allows for the combination of different knowledge sources in a uniform model, bringing mutual benefits to components. For instance, the dialogue processing module can run without any situation assessment, but its disambiguation routines can transparently benefit from it when available (since richer symbolic descriptions of objects are then available).

We want to underline the shift of focus brought by this approach during the design and integration phases of robots: components of our deliberative layer are defined and bound together by the knowledge they produce and consume. Human-robot interaction, because it supposes operations at human level and in environments with complex semantics, acts here as a motivational force.

Limits of disambiguation at semantic level. Interaction with humans implies the ability to deal with semantics: semantics of verbal interaction, semantics of gestures, etc. As a consequence, it also implies to deal with semantic disambiguation.

We have studied a prototypical example of semantic disambiguation in [17] with the children's "spygame": two players are facing each other with a set of random objects in-between, one player mentally choose one object, and the other player has to guess the object by asking closed questions like *Is your object small or large?* Based on the knowledge it has acquired, the robot is able to minimise the number of questions required to find the object.

When playing this kind of game, however, the issue arises that the robot has no way to select which knowledge about the object is relevant in the interaction context. For instance, the knowledge base may store facts like ⟨`OBJ1 type ActiveConcept`⟩ (which internally means that this concept was mentioned in a discussion in the last few seconds): this information is not a relevant property of `OBJ1` when trying to disambiguate concepts with humans. This distinction between *internal knowledge* (meaningful to the system only) and *common knowledge* (whose meaning is understood by all the agents) has not been properly dealt with in our architecture.

Besides, even knowledge that belongs to the *common knowledge* may not be appropriate in a given interaction context. For instance, the system may compute that at a given instant the human is looking at the object: ⟨`HUMAN looksAt OBJ1`⟩. This property makes sense to both parties, but in the context of the *spygame*, we would like to mainly use immanent properties, not volatile like a gaze. More research is required to identify relevant interaction contexts and knowledge classes attached to them.

6. Conclusion

6.1. A Deliberative Architecture for Social Robots

We have presented in this paper an instance of a complete deliberative architecture designed for social robots. While most of its sub-components have been independently presented in other publications, we offer here for the first time a perspective on the model of integration of these components into a coherent and consistent system for social human-robot interaction. We have first exposed our underlying knowledge model based on Description Logics [28] and some of the resulting reasoning capabilities pertaining to disambiguation [17] and mental modelling [45] that are shown to effectively scaffold interaction using human-level semantics and cognitive skills. We have then presented our approach to symbol grounding, build on an amodal situation assessment environment [61] that supports perspective taking [64, 67]. We combine it further with a situated natural language processor [74] to provide complete multi-modal interactive communication. The paper also covers our symbolic social task planner [75, 76, 77]: it generates predictive plans of the human actions that enable the system to plan for joint human-robot tasks. It can also make use of social heuristics to optimise plans for social acceptability. We briefly mention our human-aware motion and manipulation planner [91, 105, 92, 97, 93, 95], and finally present two execution controller, the PRS-based SHARY [83, 89, 2] and the event-driven PYROBOTS [85].

The integration of these components in a consistent, working and observable system builds upon the particular design of the interfaces between the cognitive components: the information streams use high-level semantics, represented as first-order logic statements. In that sense, our deliberative architecture is similar to projects like CRAM [32], KeJia [35] or PEIS Ecology [106, 39], with however a stronger emphasise on the specificities of the interaction with humans. Importantly, we distinguish ourselves from research on *cognitive architectures*: cognitive architectures are usually understood as an artificial yet principled model of (human) cognition. While some have been deployed on autonomous robots, like HAMMER [107] or ACT-R/E [22], most are primarily concerned with the modeling of human cognition and are less focused on the effective deployment on socially interactive robots. In that sense, our contribution in terms of architecture is a practical one: our integration model enables to consistently combine a large set of technically independent yet cognitively interdependent cognitive processes. We bridge them through explicit, human-level semantics, and we show that this results in a fully implemented system, effectively deployed on several platforms and in several real interaction scenarios.

This work introduces several new contributions related to the representation and the management of humans in an autonomous robotic system. Specifically, we mentioned in the introduction the following four points:

- our system achieves multi-modal and interactive grounding in complex real environments involving one or several humans and a robot;

- it supports a distributed computation of symbolic knowledge for situated dialogue, thanks to the combination of perspective taking, affordances computation and logical inference;
- it provides generic mechanisms for the robot to reason about the mental state of its human partners;
- and, by reusing the computed affordances and inference, it generates, monitors and takes part to human-robot shared plans;

While several contributions in the literature provide insights and contributions on one aspect or another (references are in the corresponding subsections), we are not aware of a fully implemented architecture that effectively combines in a coherent manner all these points. The novelty and relevance of this contribution to HRI is further underlined by the range of multi-disciplinary collaborations and studies that have been made possible by our architecture [15, 16, 17, 18, 19, 20].

6.2. The Next Steps

The design choices and the results presented here are still preliminary. While the general scheme we propose might be difficult to implement in a general sense, we believe that it is a reasonable challenge to implement it in the case of a personal robot assistant essentially devoted interactive manipulation tasks and associated activities.

One direction that we would like to further investigate is how to account for situations where divergent beliefs appear between the human and the robot. Some preliminary results have been presented in [45, 108] where we consider divergent beliefs about the state of the world, and in [13] where the robot is able to deal with divergent beliefs related to the state of the task.

There is also extensive work to be done in order to refine the notion of "good shared plan" and "good/acceptable robot behaviour" in this context. There are large avenues for learning and adaptation in this context.

Another direction to head to deals with context representation. Contexts are currently often limited to the current spatial and temporal situation. Some of our models offer the possibility to jump in the past or to switch to another agent's perspective, but in our current approach, selecting a context essentially consists in retrieving a set of beliefs corresponding to a situation, and temporarily replacing the current beliefs by those other ones. This misses the fact that at a given moment, not one but many contexts co-exist at different scales. We do not want to retrieve one monolithic set of beliefs, but instead carefully craft a context from several *atomic* contexts. Techniques for representation of overlapping "pools" of knowledge largely remain to be developed, as well as efficient algorithms to retrieve (or discard) such context-related pools of knowledge. This is a challenge not only for robotics, but more generally for artificial intelligence. The ability to explicitly manage contexts and context switches would endow the robot with a cognitive capability similar to what is known as *context-dependent memory* in cognitive psychology. This is also related to Tulving's *autonoetic consciousness* [109]: the ability to reflect upon its own past or future experiences. Much remain to be done to this regard, starting with a formal analysis of what are the relevant contexts for our robots.

Human-Robot Interaction is and will remain a challenging field for Artificial Intelligence. We hope that this contribution helps with clarifying these challenges and making them concrete decisional problems.

Acknowledgements

Building such a robotic architecture is the work of many hands, and we would like to acknowledge here the worthwhile contributions of Samir Alili, Raquel Ros Espinoza, Mamoun Gharbi, Julien Guitton, Matthieu Herrb, Jim Mainprice and Amit Kumar Pandey.

This work has been partially supported by EU FP7 CHRIS (grant 215805) and SAPHARI (grant ICT-287513) projects, and the EU H2020 Marie Sklodowska-Curie Actions project DoRoThy (grant 657227).

[1] R. Alami, On human models for collaborative robots, in: Collaboration Technologies and Systems (CTS), 2013 International Conference on, IEEE, 2013, pp. 191–194.

[2] A. Clodic, R. Alami, R. Chatila, Key elements for human-robot joint action, Sociable Robots and the Future of Social Relations: Proceedings of Robo-Philosophy 2014 273 (2014) 23–33.

[3] J. Mainprice, M. Gharbi, T. Siméon, R. Alami, Sharing effort in planning human-robot handover tasks, in: RO-MAN, 2012.

[4] J. Waldhart, M. Gharbi, R. Alami, Planning handovers involving humans and robots in constrained environment, in: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, IEEE, 2015, pp. 6473–6478.

[5] N. Sebanz, H. Bekkering, G. Knoblich, Joint action: bodies and minds moving together, Trends in cognitive sciences 10 (2) (2006) 70–76.
[6] G. Knoblich, S. Butterfill, N. Sebanz, Psychological research on joint action: theory and data, Psychology of Learning and Motivation-Advances in Research and Theory 54 (2011) 59.
[7] E. Pacherie, The phenomenology of joint action: Self-agency vs. joint-agency, Joint attention: New developments (2012) 343–389.
[8] C. Vesper, S. Butterfill, G. Knoblich, N. Sebanz, A minimal architecture for joint action, Neural Networks 23 (8) (2010) 998–1003.
[9] G. Klein, J. M. Woods, D. D.and Bradshaw, P. J. Hoffman, R. R.and Feltovich, Ten challenges for making automation a "team player" in joint human-agent activity, IEEE Intelligent Systems 19 (6) (2004) 91–95.
[10] R. Alami, M. Warnier, J. Guitton, S. Lemaignan, E. A. Sisbot, When the robot considers the human..., in: Proceedings of the 15th international symposium on robotics research, 2011.
[11] S. Harnad, The symbol grounding problem, Phys. D 42 (1-3) (1990) 335–346. doi:10.1016/0167-2789(90)90087-6.
[12] M. Fiore, A. Clodic, R. Alami, On planning and task achievement modalities for human-robot collaboration, in: Experimental Robotics, Springer International Publishing, 2016, pp. 293–306.
[13] S. Devin, R. Alami, An implemented theory of mind to improve human-robot shared plans execution, in: The Eleventh ACM/IEEE International Conference on Human Robot Interation, IEEE Press, 2016, pp. 319–326.
[14] G. Milliez, R. Lallement, M. Fiore, R. Alami, Using human knowledge awareness to adapt collaborative plan generation, explanation and monitoring, in: The Eleventh ACM/IEEE International Conference on Human Robot Interation, IEEE Press, 2016, pp. 43–50.
[15] K. Dautenhahn, M. Walters, S. Woods, K. L. Koay, C. L. Nehaniv, A. Sisbot, R. Alami, T. Siméon, How may I serve you?: a robot companion approaching a seated person in a helping context, in: Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction, ACM, 2006, pp. 172–179.
[16] K. L. Koay, E. A. Sisbot, D. S. Syrdal, M. L. Walters, K. Dautenhahn, R. Alami, Exploratory Study of a Robot Approaching a Person in the Context of Handing Over an Object, in: AAAI spring symposium: Multidisciplinary Collaboration for Socially Assistive Robotics, 2007, pp. 18–24.
[17] R. Ros, S. Lemaignan, E. A. Sisbot, R. Alami, J. Steinwender, K. Hamann, F. Warneken, Which one? grounding the referent based on efficient human-robot interaction, in: 19th IEEE International Symposium in Robot and Human Interactive Communication, 2010.
[18] F. Dehais, E. A. Sisbot, R. Alami, M. Causse, Physiological and subjective evaluation of a human–robot object hand-over task, Applied Ergonomics 42 (6) (2011) 785–791.
[19] E. Ferreira, G. Milliez, F. Lefèvre, R. Alami, Users' belief awareness in reinforcement learning-based situated human–robot dialogue management, in: Natural Language Dialog Systems and Intelligent Assistants, Springer International Publishing, 2015, pp. 73–86.
[20] M. Gharbi, P. V. Paubel, A. Clodic, O. Carreras, R. Alami, J. M. Cellier, Toward a better understanding of the communication cues involved in a human-robot object transfer, in: Robot and Human Interactive Communication (RO-MAN), 2015 24th IEEE International Symposium on, 2015, pp. 319–324. doi:10.1109/ROMAN.2015.7333626.
[21] T. W. Fong, I. Nourbakhsh, R. Ambrose, R. Simmons, A. C. Schultz, J. Scholtz, The peer-to-peer human-robot interaction project, in: Proceedings of the AIAA Space 2005, American Institute of Aeronautics and Astronautics, American Institute of Aeronautics and Astronautics, Long Beach, 2005.
[22] G. Trafton, L. Hiatt, A. Harrison, F. Tamborello, S. Khemlani, A. Schultz, ACT-R/E: An embodied cognitive architecture for human-robot interaction, Journal of Human-Robot Interaction 2 (1) (2013) 30–55.
[23] R. Alami, R. Chatila, S. Fleury, M. Ghallab, F. Ingrand, An architecture for autonomy, I. J. Robotic Res. 17 (4) (1998) 315–337. doi:10.1177/027836499801700402.
[24] E. Gat, R. P. Bonnasso, R. Murphy, On three-layer architectures, Artificial intelligence and mobile robots (1998) 195–210.
[25] R. Volpe, I. Nesnas, T. Estlin, D. Mutz, R. Petras, H. Das, The CLARAty architecture for robotic autonomy, in: Aerospace Conference, 2001, IEEE Proceedings., Vol. 1, 2001, pp. 1/121–1/132 vol.1. doi:10.1109/AERO.2001.931701.
[26] D. Goldberg, V. Cicirello, M. B. Dias, R. Simmons, S. Smith, T. Smith, A. T. Stentz, A distributed layered architecture for mobile robot coordination: Application to space exploration, in: Proceedings of the 3rd International NASA Workshop on Planning and Scheduling for Space, 2002.
[27] M. Woolridge, Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence., Massachusetts Institute of Technology, 1999, Ch. Intelligent Agents, pp. 27–78.
[28] S. Lemaignan, R. Ros, L. Mösenlechner, R. Alami, M. Beetz, ORO, a knowledge management platform for cognitive architectures in robotics, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010.
[29] S. Lemaignan, Grounding the Interaction: Knowledge Management for Interactive Robots, 2012, Ch. The Knowledge API, pp. 161–174.
[30] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, Y. Katz, Pellet: A practical OWL-DL reasoner, Web Semantics: science, services and agents on the World Wide Web 5 (2) (2007) 51–53.
[31] N. Hawes, M. Zillich, J. Wyatt, BALT & CAST: Middleware for cognitive robotics, in: In Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication (ROMAN 2007), 2007, pp. 998–1003.
[32] M. Beetz, L. Mösenlechner, M. Tenorth, CRAM — A Cognitive Robot Abstract Machine for everyday manipulation in human environments, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010.
[33] M. Tenorth, M. Beetz, KnowRob - knowledge processing for autonomous personal robots, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009.
[34] M. Waibel, M. Beetz, J. Civera, R. D'Andrea, J. Elfring, D. Galvez-Lopez, K. Haussermann, R. Janssen, J. Montiel, A. Perzylo, et al., Roboearth, Robotics & Automation Magazine 18 (2) (2011) 69–82.
[35] X. Chen, J. Ji, J. Jiang, G. Jin, F. Wang, J. Xie, Developing high-level cognitive functions for service robots, in: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '10, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2010, pp. 989–996.
[36] E. Erdem, E. Aker, V. Patoglu, Answer set programming for collaborative housekeeping robotics: representation, reasoning, and execution, Intelligent Service Robotics 5 (4) (2012) 275–291.
[37] S. Lemaignan, P. Dillenbourg, Mutual modelling in robotics: Inspirations for the next steps, in: Proceedings of the 2015 ACM/IEEE

Human-Robot Interaction Conference, 2015.
[38] D. B. Lenat, R. V. Guha, K. Pittman, D. Pratt, M. Shepherd, Cyc: toward programs with common sense, Communications of the ACM 33 (8) (1990) 30–49.
[39] M. Daoutis, S. Coradeschi, A. Loutfi, Cooperative knowledge based perceptual anchoring, International Journal on Artificial Intelligence Tools 21 (03) (2012) 1250012.
[40] S. Coradeschi, A. Saffiotti, An introduction to the anchoring problem, Robotics and Autonomous Systems 43 (2-3) (2003) 85–96. doi:10.1016/S0921-8890(03)00021-6.
[41] D. Premack, G. Woodruff, Does the chimpanzee have a theory of mind?, Behavioral and Brain sciences 1 (4) (1978) 515–526.
[42] J. Perner, J. Roessler, From infants' to children's appreciation of belief, Trends in cognitive sciences 16 (10) (2012) 519–525.
[43] H. Wimmer, J. Perner, Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception, Cognition 13 (1) (1983) 103–128.
[44] S. Baron-Cohen, A. Leslie, U. Frith, Does the autistic child have a "theory of mind" ?, Cognition.
[45] M. Warnier, J. Guitton, S. Lemaignan, R. Alami, When the robot puts itself in your shoes. managing and exploiting human and robot beliefs, in: Proceedings of the 21th IEEE International Symposium in Robot and Human Interactive Communication, 2012.
[46] C. Breazeal, J. Gray, M. Berlin, An embodied cognition approach to mindreading skills for socially intelligent robots, The International Journal of Robotics Research 28 (5) (2009) 656–680.
[47] R. Atkinson, R. Shiffrin, Human memory: A proposed system and its control processes, The psychology of learning and motivation: Advances in research and theory 2 (1968) 89–195.
[48] J. Anderson, Language, Memory, and Thought, Lawrence Erlbaum, 1976.
[49] E. Tulving, How many memory systems are there?, American Psychologist 40 (4) (1985) 385.
[50] A. Baddeley, Working memory, Current Biology 20 (4) (2010) R136–R140.
[51] J. Lehman, J. Laird, P. Rosenbloom, A gentle introduction to SOAR, an architecture for human cognition: 2006 update, Tech. rep., University of Michigan (2006).
[52] S. Shapiro, J. Bona, The GLAIR cognitive architecture, in: AAAI Fall Symposium Series, 2009.
[53] P. Baxter, J. de Greeff, R. Wood, T. Belpaeme, Modelling concept prototype competencies using a developmental memory model, Paladyn 3 (4) (2012) 200–208. doi:10.2478/s13230-013-0105-9.
[54] H. Jacobsson, N. Hawes, G.-J. Kruijff, J. Wyatt, Crossmodal content binding in information-processing architectures, in: Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction, ACM, New York, NY, USA, 2008, pp. 81–88. doi:10.1145/1349822.1349834.
[55] N. Mavridis, D. Roy, Grounded situation models for robots: Where words and percepts meet, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006.
[56] A. Nüchter, J. Hertzberg, Towards semantic maps for mobile robots, Robotics and Autonomous Systems 56 (11) (2008) 915 – 926. doi:10.1016/j.robot.2008.08.001.
[57] C. Galindo, J. Fernández-Madrigal, J. González, A. Saffiotti, Robot task planning using semantic maps, Robotics and Autonomous Systems 56 (11) (2008) 955–966.
[58] N. Blodow, L. C. Goron, Z.-C. Marton, D. Pangercic, T. Rühr, M. Tenorth, M. Beetz, Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments, in: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 2011.
[59] C. Lörken, J. Hertzberg, Grounding planning operators by affordances, in: International Conference on Cognitive Systems (CogSys), 2008, pp. 79–84.
[60] K. Varadarajan, M. Vincze, Ontological knowledge management framework for grasping and manipulation, in: IROS Workshop: Knowledge Representation for Autonomous Robots, 2011.
[61] E. Sisbot, R. Ros, R. Alami, Situation assessment for human-robot interaction, in: 20th IEEE International Symposium in Robot and Human Interactive Communication, 2011.
[62] J. McCarthy, From here to human-level AI, Artificial Intelligence.
[63] H. Moll, M. Tomasello, Level 1 perspective-taking at 24 months of age, British Journal of Developmental Psychology 24 (3) (2006) 603–614.
[64] L. Marin, E. A. Sisbot, R. Alami, Geometric tools for perspective taking for human-robot interaction, in: Mexican International Conference on Artificial Intelligence (MICAI 2008), Mexico City, Mexico, 2008.
[65] B. Tversky, P. Lee, S. Mainwaring, Why do speakers mix perspectives?, Spatial Cognition and Computation 1 (4) (1999) 399–412. doi:10.1023/A:1010091730257.
[66] C. Breazeal, M. Berlin, A. Brooks, J. Gray, A. Thomaz, Using perspective taking to learn from ambiguous demonstrations, Robotics and Autonomous Systems (2006) 385–393.
[67] R. Ros, E. A. Sisbot, R. Alami, J. Steinwender, K. Hamann, F. Warneken, Solving ambiguities with perspective taking, in: 5th ACM/IEEE International Conference on Human-Robot Interaction, 2010.
[68] J. O'Keefe, The Spatial Prepositions, MIT Press, 1999.
[69] C. Matuszek, D. Fox, K. Koscher, Following directions using statistical machine translation, in: Proceedings of the International Conference on Human-Robot Interaction, ACM Press, 2010.
[70] T. Regier, L. Carlson, Grounding spatial language in perception: An empirical and computational investgation, Journal of Experimental Psychology.
[71] J. Kelleher, G. Kruijff, Incremental generation of spatial referring expressions in situated dialog, in: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2006, pp. 1041–1048.
[72] S. N. Blisard, Modeling spatial referencing language for human-robot interaction, in: in Proc. IEEE Intl. Workshop on Robot and Human Interactive Communication, 2005, pp. 698–703.
[73] A. K. Pandey, R. Alami, Affordance graph: A framework to encode perspective taking and effort based affordances for day-to-day human-

robot interaction, in: IROS 2013, IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2013, pp. 2180–2187.
[74] S. Lemaignan, R. Ros, E. A. Sisbot, R. Alami, M. Beetz, Grounding the interaction: Anchoring situated discourse in everyday human-robot interaction, International Journal of Social Robotics (2011) 1–19doi:10.1007/s12369-011-0123-x.
[75] S. Alili, V. Montreuil, R. Alami, HATP task planer for social behavior control in autonomous robotic systems for HRI, in: The 9th International Symposium on Distributed Autonomous Robotic Systems, 2008.
[76] S. Alili, M. Warnier, M. Ali, R. Alami, Planning and plan-execution for human-robot cooperative task achievement, in: 19th International Conference on Automated Planning and Scheduling, 2009.
[77] R. Lallement, L. De Silva, R. Alami, HATP: An HTN planner for robotics, in: Proceedings of the PlanRob 2014, ICAPS, 2014.
[78] B. J. Grosz, S. Kraus, Collaborative plans for complex group action, Artificial Intelligence 86 (1996) 269–358.
[79] H. H. Clark, Using Language, Cambridge University Press, 1996.
[80] C. Kemp, E. Edsinger, A. Torres-Jara, Challenges for robot manipulation in human environments, Robotics & Automation Magazine.
[81] S. Lemaignan, Grounding the interaction: Knowledge management for interactive robots, Ph.D. thesis, CNRS - Laboratoire d'Analyse et d'Architecture des Systèmes, Technische Universität München - Intelligent Autonomous Systems lab (2012).
[82] S. Lallée, S. Lemaignan, A. Lenz, C. Melhuish, L. Natale, S. Skachek, T. van Der Zant, F. Warneken, P. Dominey, Towards a platform-independent cooperative human-robot interaction system: I. perception, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 4444–4451. doi:10.1109/IROS.2010.5652697.
[83] A. Clodic, H. Cao, S. Alili, V. Montreuil, R. Alami, R. Chatila, SHARY: A Supervision System Adapted to Human-Robot Interaction, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 229–238. doi:10.1007/978-3-642-00196-3_27.
[84] F. Ingrand, R. Chatila, R. Alami, F. Robert, PRS: A high level supervision and control language for autonomous mobile robots, in: Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 1, 1996, pp. 43–49.
[85] S. Lemaignan, A. Hosseini, P. Dillenbourg, pyRobots: a toolset for robot executive control, in: Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2015.
[86] C. Rich, C. L. Sidner, Collagen: When agents collaborate with people, Proceedings of the first international conference on Autonomous Agents.
[87] C. L. Sidner, C. Lee, C. Kidd, N. Lesh, C. Rich, Explorations in engagement for humans and robots, Artificial Intelligence 166 (1-2) (2005) 140–164.
[88] A. Clodic, M. Ransan, R. Alami, V. Montreuil, A management of mutual belief for human-robot interaction, in: 2007 IEEE International Conference on Systems, Man and Cybernetics, 2007, pp. 1551–1556. doi:10.1109/ICSMC.2007.4414019.
[89] M. Fiore, A. Clodic, R. Alami, On planning and task achievement modalities for human-robot collaboration, in: 14th International Symposium on Experimental Robotics, 2014.
[90] J. Austin, J. Urmson, M. Sbisà, How to do things with words, Harvard University Press, 1962.
[91] E. Sisbot, L. Marin-Urias, R. Alami, T. Siméon, A human aware mobile robot motion planner, IEEE Transactions on Robotics 23 (5) (2007) 874–883.
[92] J. Mainprice, E. A. Sisbot, L. Jaillet, J. Cortes, R. Alami, T. Simeon, Planning human-aware motions using a sampling-based costmap planner, in: IEEE International Conference on Robotics and Automation, 2011.
[93] E. A. Sisbot, R. Alami, A human-aware manipulation planner, IEEE Transactions on Robotics 28 (5) (2012) 1045–1057.
[94] K. Madhava, R. Alami, T. Simeon, Safe proactive plans and their execution, Robotics and Autonomous Systems 54 (3) (2006) 244–255.
[95] T. Kruse, A. K. Pandey, R. Alami, A. Kirsch, Human-aware robot navigation: A survey, Robotics and Autonomous Systems 61 (12) (2013) 1726–1743.
[96] J. Rios-Martinez, A. Spalanzani, C. Laugier, From proxemics theory to socially-aware navigation: A survey, I. J. Social Robotics 7 (2) (2015) 137–153. doi:10.1007/s12369-014-0251-1.
[97] A. Pandey, M. Ali, M. Warnier, R. Alami, Towards multi-state visuo-spatial reasoning based proactive human-robot interaction, in: Proceedings of the 15th International Conference on Advanced Robotics, IEEE, 2011, pp. 143–149.
[98] S. Lemaignan, R. Ros, R. Alami, M. Beetz, What are you talking about? grounding dialogue in a perspective-aware robotic architecture, in: Proceedings of the 20th IEEE International Symposium in Robot and Human Interactive Communication, 2011.
[99] S. Lemaignan, M. Gharbi, J. Mainprice, M. Herrb, R. Alami, Roboscopie: A theatre performance for a human and a robot, in: Proceedings of the 2012 Human-Robot Interaction Conference, 2012.
[100] R. Alami, M. Warnier, J. Guitton, S. Lemaignan, E. A. Sisbot, When the robot considers the human..., in: Proceedings of the 15th International Symposium on Robotics Research, 2011.
[101] M. Gharbi, S. Lemaignan, J. Mainprice, R. Alami, Natural interaction for object hand-over, 2013.
[102] S. Lemaignan, R. Alami, Talking to my robot: from knowledge grounding to dialogue processing, in: Proceedings of the 2013 Human-Robot Interaction Conference, 2013.
[103] F. Varela, E. Thompson, E. Rosch, The embodied mind: Cognitive science and human experience, The MIT Press, 1992.
[104] R. Pfeifer, J. Bongard, How the body shapes the way we think: a new view of intelligence, MIT press, 2007.
[105] E. A. Sisbot, A. Clodic, R. Alami, M. Ransan, Supervision and motion planning for a mobile manipulator interacting with humans, in: Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction, 2008.
[106] A. Saffiotti, M. Broxvall, Peis ecologies: Ambient intelligence meets autonomous robotics, in: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies, ACM, 2005, pp. 277–281.
[107] Y. Demiris, B. Khadhouri, Hierarchical attentive multiple models for execution and recognition of actions, Robotics and autonomous systems 54 (5) (2006) 361–369.
[108] G. Milliez, M. Warnier, A. Clodic, R. Alami, A framework for endowing an interactive robot with reasoning capabilities about perspective-taking and belief management, in: The 23rd IEEE International Symposium on Robot and Human Interactive Communication, IEEE, 2014, pp. 1103–1109.
[109] E. Tulving, Memory and consciousness, Canadian Psychology/Psychologie Canadienne 26 (1) (1985) 1.