Int J of Soc Robotics (2014) 6:593–620
DOI 10.1007/s12369-014-0246-y

Towards Human-Level Semantics Understanding of Human-Centered Object Manipulation Tasks for HRI: Reasoning About Effect, Ability, Effort and Perspective Taking

Amit Kumar Pandey · Rachid Alami

Accepted: 31 May 2014 / Published online: 14 July 2014

Abstract In its lifetime, a robot should be able to autonomously understand the semantics of different tasks to effectively perform them in different situations. In this context, it is important to distinguish the *meaning* (in terms of the desired effect) of a task and the *means* to achieve that task. Our focus is those tasks in which one agent is required to perform a task for another agent, such as *give*, *show*, *hide*, *make-accessible*, etc. In this paper, we identify that a high-level human-centered combined reasoning, based on perspective taking, efforts and abilities analyses, is the key to understand semantics of such tasks. By combining these aspects, the robot infers sets of hierarchy of facts, which serve for analyzing the effect of a task. We adapt the explanation based learning approach enabling the task understanding from the very first demonstration and continuous refinement with new demonstrations. We argue that such symbolic level understanding of a task, which is not bound to trajectory, kinematics structure or shape of the robot, facilitates generalization to novel situations as well as ease the transfer of acquired knowledge among heterogeneous robots. Further, the knowledge of tasks at such human understandable level of abstraction will enrich the natural human–robot interaction.

Keywords Perspective taking · Ability analysis · Effort analysis · Emulation learning · Effect understanding · Social learning · Explanation based learning · Task semantics

1 Introduction

As robots are evolving to co-exist in human centered environment, they will be expected to understand as well as refine their knowledge about various tasks in day-to-day life. Further, such understandings should enable them to perform those tasks in heterogeneous situations without any need of providing the learning data for each and every situation.

From the perspective of social learning, which in loose sense is "A observes B and then 'acts' like B", in [11], three components have been identified: Goal, Action and Result. Based on what is learned there are basically three categories: *mimicking*, *emulation* and *imitation*. Mimicking is just reproducing the action without any goal. *Emulation* is regarded as learning causal properties of objects [61], and learning the goal by observation [45,64], to bring the same result, which might be by different means/actions than the demonstrated one. *Imitation* [40,54] is bringing the same result and with same actions. Therefore, *emulation* involves reproducing the changes in the state of the environment that are the results of the demonstrator's behavior, whereas *imitation* involves reproducing the actions that produced those changes in the environment.

Emulation is regarded as an important social learning skill, also among great apes [61] and children [29,45]. In fact, this facilitates to perform same task in different ways. Some components and evidence of successful *emulation* during inter-

This work is partially conducted within Romeo2 (Humanoid Robot Assistant and Companion for Everyday Life) project (http://projetromeo.com/), Funded by BPIFrance in the framework of the Structuring Projects of Competitiveness Clusters (PSPC).

A. K. Pandey (✉)
A-Lab, Aldebaran Robotics, 168 bis 170 rue Raymond Losserand,
75014 Paris, France
e-mail: akpandey@aldebaran-robotics.com

R. Alami
CNRS, LAAS, 7 avenue du colonel Roche,
F-31400 Toulouse, France
e-mail: rachid.alami@laas.fr

R. Alami
Université de Toulouse, LAAS, F-31400 Toulouse, France

action can be traced even at the child's early development phase. Children are able to show an object to someone in different ways [37]: by pointing, by turning the object, by holding it so that other can see it. Similarly, they are able to hide an object from another person in different ways [25]: by placing a screen between the person and the object, by placing the object itself behind the screen from the person's perspective. All these suggest that we are able to abstract the '*desired effect*' to bring, that too by reasoning from the other person's perspective. In our discussion above, such desired effect were: the object should be visible to the other person for the task of showing; the object should be invisible from the other person's perspective for the task of hiding. Therefore, for such tasks to be successfully achieved, as a sign of *emulation*, i.e. bringing the same result, which might be with different means/actions than the demonstrated one, understanding the desired effect and reasoning about perspective are important.

Motivated from these evidences, we also separate *emulation* and *imitation* parts during *learning by demonstration* for the robots. In this paper, we will develop a framework for human-level understanding of effect-based semantics of a task, independent from its execution. Such 'meaningful' understanding will also provide flexibility of planning alternatively for a task depending upon the situation, in addition to enriching the natural human–robot interaction.

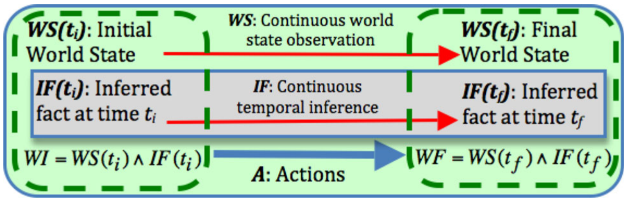

WS(t_i): Initial World State
WS: Continuous world state observation
WS(t_f): Final World State
IF(t_i): Inferred fact at time t_i
IF: Continuous temporal inference
IF(t_f): Inferred fact at time t_f
WI = WS(t_i) ∧ IF(t_i)
A: Actions
WF = WS(t_f) ∧ IF(t_f)

Fig. 1 $\langle WI, A, WF \rangle$ triplet, showing Causal Nature of Environment Change, a set of actions A on initial world *WI* at time t_i results into a final world *WF* at time t_f

We hypothesize a task tk as a series of actions A by a set of agents Ag, causing change in the world state from $WS(t_i)$ to $WS(t_f)$, similar to [42]. Further, we assume that various facts are inferred continuously during a course of action, and $IF(t)$ represents the facts inferred at time stamp t. Hence, by observing and analyzing an instance of $<WI, A, WF>$, (see Fig. 1) various parameters of a task such as preference, desired and undesired changes, trajectory, etc. could be learned. Where $WI = (WS(t_i), IF(t_i))$ and $WF = (WS(t_f), IF(t_f))$, t_i and t_f are the time stamps corresponding to start and end of A. Here it is important to note that *WI* and *WF* are the snapshots at a particular instant, however, as the inference of facts is continuous, it facilitate to assert dynamic and static aspects of the environment, for example "box on table", "ball moving", etc. Further, depending upon the level of abstraction, A could be symbolically described as a single action or a series of actions.

Table 1, gives a general idea about the possible components, which could be inferred and learned by partially or fully observing different components of <*WI*, *A*, *WF*> tuple. In this paper our focus is at <*WI*, *WF*> level, marked as * in Table 1, by reasoning about the *effect*, assuming positive examples.

Human-Centered Object Manipulation Tasks (HCOM Tasks) We define the tasks within the context of the paper as *human-centered object manipulation tasks*. These are the tasks in which one agent (robot or human) is performing some object manipulation task for another person, by considering his/her presence and reasoning about him/her. Hence, such object manipulation tasks require human-centered reasoning, beyond the reasoning about stability of grasp, placement, etc. from an object-centered perspective.

The focus of this paper is the subset of such *human-centered object manipulation tasks*, which requires *pick, place, hold* like actions. Therefore, next we will first present the related works in learning *pick-and-place* type tasks, which addresses one or the other aspect of Table 1, followed

Table 1 Observation and learning components correlation

Information about a task T	What could be learnt for task T
WI	Preconditions, initial world state
A	Trajectory, sub-actions
WF	Post-conditions, desired world state
WI, A	Trajectory, sub-actions, initial world state based preferences about selection of actions, sub-actions and trajectory.
A, WF	Trajectory, sub-actions, desired world state based preferences about which, how and where for actions, sub-actions and trajectories to perform T
WI, WF	*Effect based semantics**, desired, undesired and side effects and changes, preferences about where to perform ***T***
WI, A, WF	Preconditions, effects, initial situation and desired world state based preferences about which, how and where for actions, sub-actions and trajectories to perform ***T***

* This effect based task understanding (important for emulation learning) will be one of the contributions of the paper

by related works on effect based learning approaches. Then we will precise our scope and discuss our motivation to have the hierarchies of facts, which enable comparing as well as qualifying the effects of a task to be learned. Followed by this, the instantiation of hierarchical knowledgebase for our domain will be presented. Then we will present the learning framework, followed by experimental results and analyses for different tasks. The conclusion will be preceded by discussion on potential applications and benefits of such human-level understanding of the tasks.

2 Related Work

Different learning aspects of Table 1 have been addressed by researchers, see [4] for a survey. At the level of trajectory, in [28], the robot learns to *pick-and-place* with constraints on orientations. In [44], the task of pouring by a human performer has been adapted by the robot for maintaining collision free movement. Similarly, other works adapt the learned trajectory for modified scenarios [7,21]. Whereas, in [66], trajectory is learned from the perspective of maintaining critical aspects of the motion. Such approaches are in fact complementary to learning the symbolic description of the task: *what does the task mean?*, which will facilitate how (at non trajectory level) to perform the task in different ways in various situations.

At symbolic primitives level, the task is mainly learned in two forms: (1) based on the sequence of sub-actions and (2) based on the effect in terms of changes in the environment. In the sub-action learning approaches the task of '*place an object next to another object*' would be inferred as '*reach*', '*grasp*' and '*transfer_relative*' [14]. The task of '*Take a bottle out of the fridge*' would be sub-symbolized as '*Open the fridge*', '*Grasp the bottle*', '*Get the bottle out*', '*Close the fridge*' and '*Put the bottle on the table in a stable position*' [20]. In [52], incremental learning of the task precedence graph, for the tasks of *pouring a bottle* and *laying a table*, have been presented. In [33], the robot grounds the *table assembling* task in terms of '*reach*', '*pick*', '*place*' and '*withdraw*', and tries to learn the dependencies in order to reorder and adapt for different initial setups. In [46], a hybrid approach tries to represent the entire task in a symbolic sub-task manner but also incorporates trajectory information to perform the task. In [38], learning to *assemble an electric switch* is presented, by providing an example of the plan by moving the manipulator through the desired assembly. The system observes the assembly and infers the underlying plan. In [16], the authors abstract the human demonstration of *assembly construction* to a sequence of object connections, which are used to infer a motion grammar for the robot to repeat the task. In [53], a probabilistic approach to learn relational planning rules has been presented, based on state representation by facts like *on*, *clear*, etc., and actions like *paint*, *pickup*, *drive*, *walk*, etc., covering different domains.

However, most of these approaches actually reason on *actions*, i.e. trying to represent a task in sub-tasks/sub-actions from the point of view of execution, which in fact is to facilitate *imitation* aspect of learning as discussed earlier. There is no explicit reasoning about the semantics of the task independent of the execution. As mentioned earlier our focus will be on task understanding from the effect point of view, to facilitate *emulation* learning. In fact, recognizing the effect of actions, based on initial and resulting world states, is an important component of *causal learnability*, and a complementary aspect for reasoning at action level, i.e. how to generate that effect [42].

For successful *emulation*, the robot should be able to analyze the effects in terms of the task driven changes. In [9], through dialogue, the task '*to follow*' a person would be understood as *to remain within 1 meter* of the person. From the perspective of learning object manipulation tasks by observing human demonstrations, in [22] the effect of *pick-and-place* type tasks have been analyzed by using predicates such as *holding object*, *hand empty*, *object at location*, etc. In [43] the robot performs different actions such as grasp, touch and tap on different objects to analyze the effects; once learned could be used to select the appropriate action for achieving a particular effect [39]. However, the effects of each action on the object were described in terms of *velocity*, *contact* and *object-hand distance*. In [60], a first order knowledge representation and processing system KnowRob, represents the knowledge in action centric way and learns the action models of real world *pick-and-place* type task domain, coupled with object and its properties. In [58] an approach has been presented to learn abstract level action selection from observation by considering the *position*, *orientation*, and the symbolic interpretations of the performer's body movement, such as *bow* and *pick object*. In [13], the robot grounds the goal of the observed tasks by symbolic concepts like '*on the shelf*', '*left side of the table*', etc. for *pick-and-place* type tasks on a table top. In [31], planning models for dexterous tasks, such as *push a slider*, are learned based on automatically generated contact constraints and automatically relaxing them whenever necessary, by addressing the *correspondence problem* because of the structural differences between the robot and the human.

However, as the focus of these works are not those types of tasks, which require one agent to perform for another agent, hence, these works does not exploit and address the necessary aspect of reasoning based on perspective taking, abilities and effort of the *target-agent*'s (the agent for whom the task is being performed).

In this paper we will focus on such human-centered object manipulation tasks, which make it a must to reason on the effects from the perspective of changes in *target-agent*'s abilities and effort, hence requiring to develop a learning framework based on the combined reasoning of such aspects.

3 Scope, Motivation and Contributions

We will consider a set of basic yet key *human-centered object-manipulation tasks* in a typical interaction scenario such as *give*, *make accessible*, *show*, *hide*, *put-away*, *hide-away* an object. One common effect of such tasks is to enable and/or disable the actions or abilities of the *target-agent* (the agent for whom the task is being performed). Such as *make accessible* enables the *target-agent* to take the object whenever he/she wants. *Hide* deprives the *target-agent* from the ability to see the object. Hence, the reasoning about the effect of a task from the *target-agent*'s perspective is a must for understanding such tasks.

3.1 An Illustrative Example

Let us consider an example scenario of making an object accessible to a person, i.e. an object, which is currently invisible and/or unreachable for a person, should be made visible and reachable to him/her. In Fig. 2 person *P1* has to make the green bottle (marked by red arrow) accessible to person *P2*. The task is same, however depending upon the current mental and physical states and desires of both them and their relation, *P1* could prefer to perform the task by choosing to put the bottle at different places, (b), (c) and (d). Here, the interesting point is, *P1* is able to infer from *P2*'s perspective that if *P2* will stand up, lean forward, and stretch out her arm, she can get the bottle in (b), whereas in (c), *P2* will be just required to stretch out her arm. In (d) as an attempt to balance mutual efforts, *P1* leans forward and puts the bottle at a place, which requires *P2* also to lean and stretch out the arm to take it. This suggests that the robot should be also able to perform the perspective taking not only from the current state but also from different states of the agent. Hence, this shows the necessity of combined reasoning based on effort, ability and perspective taking.

Now, assume that the robot is observing the task as performed in Fig. 2c, and learns just by reasoning about the actions, in terms of symbolic sub-tasks such as *grasp object*, *carry object* and *put object* at 'x' distance from the person *P2* or put the object *reachable* by P2's *current position*. In this case, it will not be able to identify that the tasks performed in Fig. 2b, d are the same tasks. This is because of two main reasons: (1) what the robot has learned is, how to perform the task, (2) it did not reason at correct level of abstraction required for such tasks. In this example a better understanding of the task should be: *the object should become 'easier' to be seen, reached and grasped by the target-agent, P2*. Hence, the robot should be able to infer the facts at different levels of abstractions, which might not be directly observable, such as comparative facts: *easier*, *difficult*, etc. and use them in the learning process. This points towards building a hierarchy of knowledge based on different levels of abstractions.

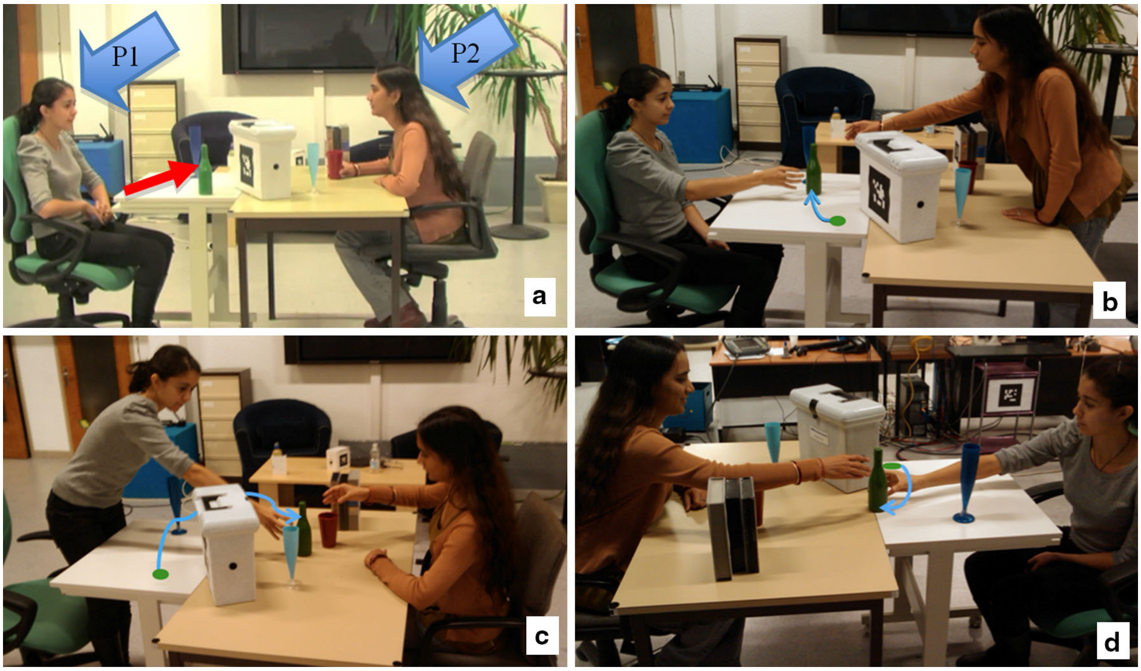


Fig. 2 **a** Initial scenario for the task of making the *green bottle* (indicated by *arrow*) accessible to *P2* by *P1*. *P1* puts the bottle so that it will be visible and reachable by *P2* if she will **b** stand up, lean forward and stretch out her arm, **c** just stretch out the arm, **d** lean forward and stretch out the arm from the sitting position. (Color figure online)

3.2 Main Contributions

In [42] two desirable capabilities of an autonomous causal learnability have been discussed as: (a) Ability to infer the indirect facts, which could be obtained by ramifications of the action's effects. (b) Build a hypothesis that the agent can use to make predictions of effect-based resultant world state from a novel initial state, which has not been observed before.

The main contributions of the paper are to deal with the above-mentioned two components by incorporating the aspects explored earlier in this section in the following manner:

(i) *Hierarchical knowledge building incorporating effort, ability and perspective taking:* Enriching the robot's knowledge with a set of hierarchy of facts related to agent's capabilities and object state. We enable the robot to infer comparative facts such as *easier*, *difficult*, *maintained*, *reduced*, etc. as well as qualitative facts such as *supportive*, *non-supportive*, etc. This requires what we call as *multi-state visuo-spatial perspective taking* reasoning about the agents, i.e. by combining the aspects of *effort*, *ability* and *perspective taking*. To the best of our knowledge, facts based on such reasoning have not been inferred and used in the context of the robot understanding tasks from demonstrations.

(ii) *Human-level task semantics understanding through explanation based learning:* We present an explanation based learning (EBL) framework to learn effect-based tasks' semantics by building a hypothesis tree. Further, we have incorporated *m-estimate* based reasoning to find consistency based relevant predicates for a task, which also incorporates the notion of *experience*. The framework autonomously learns at the appropriate levels of abstractions for different tasks. We argue that such human-level understanding successfully holds for novel scenarios as well as facilitates transfer of understanding among heterogeneous robots.

Positive Demonstrations by Expert Teacher Learning based on expert demonstrations is an acceptable practice for learning explanation and goal driven autonomy [63,65]. Therefore, we will also assume that the demonstrations are by expert teachers and we are not trying to teach something to a "child" robot with non-expert teacher or wrong demonstrations.

4 Effect-Centric Knowledge Abstractions Building: A General Guideline

As discussed, to capture the 'meaning' of the aforementioned tasks, it is important to reason about the capabilities and constraints of the agents involved. Further, to facilitate learning at appropriate abstraction level, there should be different levels of abstractions based on those facts, which might not be directly observable through sensor based data, for example the fact that something became *easier*. Hence, in this section and in the section following, we present the first contribution of the paper: hierarchical knowledge building, by enabling the robot to infer the facts at different levels of abstractions.

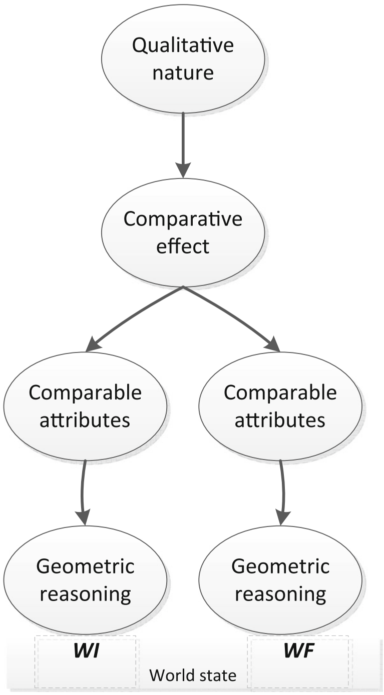


Fig. 3 A general representation of fact hierarchy from the perspective of analyzing effect

The hierarchical knowledge is built from the perspective of analyzing effects, hence requires the comparative reasoning about the initial and the final states corresponding to the task. We consider two types of effects to be incorporated in the hierarchy: (1) Ability based effects: building relations based on changes in applicability of abilities of the agent. (2) Perceptual situation based effects: building relations based on changes in perceived information about the status and the situation of the world. Our approach of developing the hierarchy is also influenced by the intuitive understanding of the tasks and their semantic differences. For example, if the task is to show something, we know that the object's visibility should *increase* for the *target-agent* (the agent to whom the object is being shown). Hence, there should not only be an attribute related to measuring visibility but also some attribute should capture the notion of *change* in the visibility.

Based on such reasoning, we proposed different levels of abstraction in the knowledgebase, see Fig. 3. (we assume level 1 is highest level of abstraction):

(i) *Level 4, geometric reasoning*: Agents' configurations and the geometric world state. At this level different types of geometric reasoning can be performed to extract the values of some geometric attributes. Those can be the geometric position of the object, reasoning about IK based actions for agents, etc.
(ii) *Level 3, comparable attributes*: Quantitative and symbolic reasoning based attributes, which can be directly used to compare two values. Based on the lower level such facts could be the visibility score of an object, the effort of an agent, the state of the object, etc.
(iii) *Level 2, comparative effect*: This corresponds to the initial and final world states based relative comparisons of the ways the value of an attribute changes or maintained. The comparable facts at level 3 are used to derive such comparative effects. In general, the comparison result could be stated in different forms, such as unchanged, easier, difficult, increased, decreased, maintained, gained, lost, etc. Therefore, the choice of range of the values of such facts depends upon the attribute itself, the context and the requirement.
(iv) *Level 1, qualitative nature*: This corresponds to the symbolic facts, which qualify the nature of the comparative effect and the resulting changes in the value of its child attribute. Existence of Such level of abstraction is more relevant, if comparative effects in the level 2 can have more than two types and some of them can be grouped together in a meaningful manner. For example, if comparative effect about visibility can have three values: increased, maintained, and decreased, then we can group and quantify first two values as of supporting nature, which support the agent's ability to see, whereas the decreased value can be seen as unsupportive nature for the agent's ability to see.

Depending upon the target domain of the task and the type of the attribute, we can decide about the different levels of abstraction to be included in building the corresponding knowledge hierarchy. Such as, if there is an obvious meaning of intention behind some changes in the effect and at least two such changes could be further assigned a single meaning, then we can have the fact that qualifies the nature of that intention. For, example, if a task maintains or better facilitates for a particular ability of an agent, we can say that there is an intention to support that ability, so we will have the abstraction up to level 1. On the other hand, if the attribute at level 2 can take only two possible values, there might be no reason to further qualify those values. For example, if something can be either changed or maintained, then there is no need to further abstract them at a higher level. In that case, any other label will just serve as synonym.

Depending upon the context and the implementation, the selection of different attributes, different ability types of the agent to consider as well as the computation of the effect can vary. Below, we describe the attributes and methods we have chosen to instantiate the hierarchical knowledgebase adapted to our domain of human-centered object manipulation (HCOM) tasks.

5 Instantiation of the Hierarchical Knowledgebase for Basic Human-Centered Object Manipulation Tasks Domain

In this section, we identify different basic attributes to instantiate the hierarchical knowledgebase for the current context of human-level understanding of the basic HCOM tasks. As discussed in Sect. 3 one contribution of the paper is to identify the importance of combined reasoning about abilities, effort and the perspective taking. In this direction, based on such combined reasoning, we have identified some important predicates. Further, we have also shown in Sect. 8.3 that without them an intuitive understanding of a task cannot be achieved.

The guidelines of the different levels of abstraction outlined in Fig. 3 will be used to instantiate different hierarchies of attributes related to the human and the object. In the trees of the hierarchical representation, the superscripts *1* and *2* represents the facts related to the initial world state *WI* and the final world state *WF*, respectively.

We have explain in the experimental platform Sect. 7.1 that the robot maintains and updates 3D world model, to represent a geometric *world state*, similar to the one shown in Fig. 4a. By reasoning on this 3D model of the world, the robot infers various facts related to agent abilities, object state and affordances. Next, we will first discuss various abilities of the agents, our robot is able to infer, and then instantiate a subset of hierarchical knowledgebase based on combined reasoning about abilities, effort and perspective taking. Then we will discuss agent and object status based hierarchy of facts and eventually build our hierarchical knowledgebase of the domain, within the scope of the paper.

5.1 Abilities, Effort and Perspective Taking Based Knowledge Building

By reasoning on the agents' models, the robot estimates various abilities of the agents. An *agent's state*, *S*, is defined by his/her/its configuration, position and orientation. Reasoning on the agent state and the environment, various abilities to see, reach and grasp are inferred.

5.1.1 Ability to See (Se) and the Associate Visibility Score (ViS)

An object is said to be seen if at least one cell belonging to the object is visible by the agent from a particular state. Further,

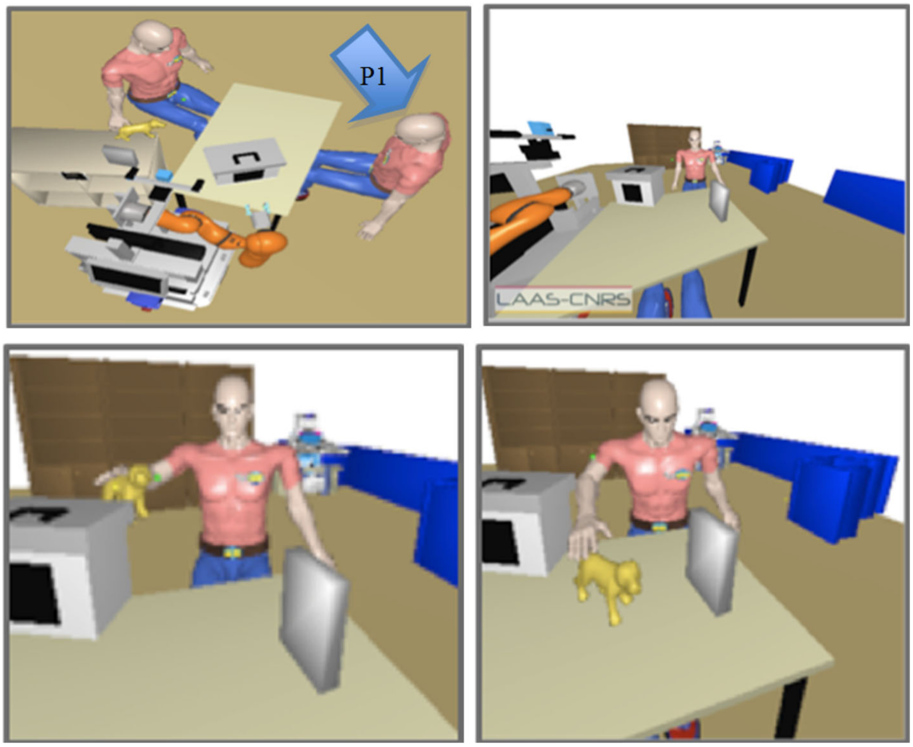


Fig. 4 **a**, **b**: First row, **c**, **d**: Second row. **a** Robot observing a human–human interaction. **b** *P1*'s current state visual perspective, visibility scores (*ViS*) is *0.0* for entirely hidden toy dog. **c** *ViS* is *0.001* when the toy dog is partially occluded and relatively far. **d** *ViS* is *0.003* for non-occluded and relatively closer toy dog

the robot keeps track of how much the object is visible. For this the robot calculates the visibility score, *ViS*, by dividing the number of pixels of the object in the image of the field of view of agent by the total number of pixels in the image of the field of view, from a particular state of the agent. Figure 4b–d shows different visibility scores for toy horse from human *P1* perspective from his current state.

5.1.2 Ability to Reach (Re)

An object is said to be reachable if at least one cell (in the grid based representation of the workspace, see [47] for implementation detail) belonging to the object is within the length of the fingertip from the shoulder in a particular state of the agent; that is how we also perceive a rough reachability [10].

5.1.3 Ability to Grasp (Gr)

As an object might be reachable to an agent for various purposes such as to touch, push, point, grasp, etc., the robot further distinguishes whether the reachable object is graspable or not. Our robot can generate a set of grasps for different multi-fingered hands/grippers, for object of different shapes [57]. Figure 5 shows a subset of the generated grasps for different objects for the robot's gripper and for an anthropomorphic hand used to reason on graspability of the human. If there exists at least one collision free grasp for the reachable object, the object is assumed to be graspable by the agent's hand.

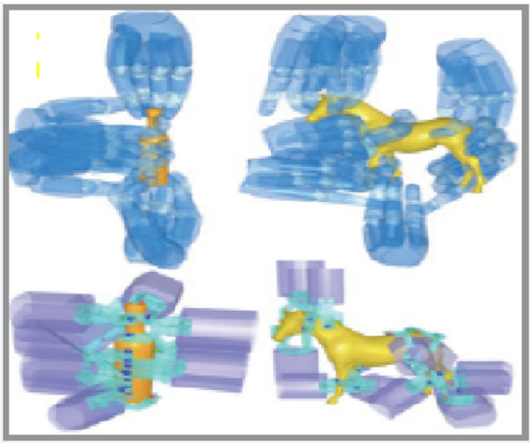


Fig. 5 Subsets of grasps for an anthropomorphic hand and the robot's gripper, generated for different objects (see [57])

5.1.4 Multi-State Perspective Taking

Perspective taking has already been shown as an important aspect in learning [6]. In fact, it has been shown as a key component in shaping and grounding our day-to-day interactions. Perspective taking is important for how we interact with others [24], to reduce ambiguity and for grounding [55,62], for action recognition [30], for proactive behavior [50], planning interactive and cooperative tasks [34], sharing attention [41] and so on.

In [47], taking inspiration from the studies in neurosciences and behavioral psychology, we have presented the concept of *Mightability*, which stands for *Might be Able to…*, based on the reasoning about multi-state visuo-spatial perspective taking. The idea is to analyze various abilities such as ability to reach, ability to see, etc. of an agent not only from the current state of the agent, but also from a set of sates, which the agent might achieve from his/her/its current state. For this the robot applies, A_V, an ordered list of virtual actions, to make the agent virtually attain a state and then estimates the abilities $Ab \in \{See, Reach, Grasp\}$, by respecting the environmental and postural constraints of the agent. Currently,

$$A_V \subseteq \left\{A_V^{head}, A_V^{arm}, A_V^{torso}, A_V^{posture}, A_V^{displace}\right\} \tag{1}$$

where,

$$A_V^{head} \subseteq \{Pan\_Head, Tilt\_Head\} \tag{2}$$

$$A_V^{arm} \subseteq \{Stretch\_Out\_Arm\,(left|right)\} \tag{3}$$

$$A_V^{torso} \subseteq \{Turn\_Torso, Lean\_Torso\} \tag{4}$$

$$A_V^{posture} \subseteq \{Make\_Standing, Make\_Sitting\} \tag{5}$$

$$A_V^{displace} \subseteq \{Move\_To\} \tag{6}$$

This enables the robot to answer following queries:

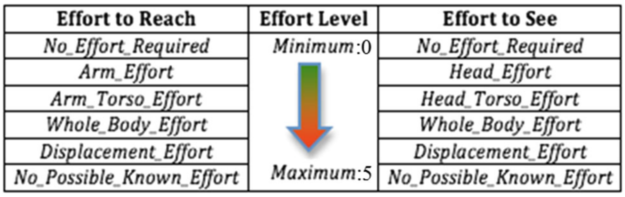
$$?\{obj\} \Rightarrow \{Ab = True|False, apply(A_v), by(Ag)\} \tag{7}$$

$$?\{A_v\} \Rightarrow \{Ab = True|False, for(obj), by(Ag)\} \tag{8}$$

For example, from Eq. 7, the robot can find the objects, which will be visible, *Ab:see=True*, if the human, *Ag=human*, will stand up and lean forward, i.e. A_v*=[Make_Standing, Lean_Torso]*. Whereas by Eq. 8, the robot could find the ordered list of action A_v required by an agent to see, reach or grasp a particular object.

5.1.5 Effort Based Comparable Facts: Types of Efforts for Agent's State Change

The robot needs to quantify the efforts associated with actions to attain a state from another state of the agent. For this, the robot associates a type to effort in terms of involved body parts for the virtual action A_v. Figure 6 shows the types, which in fact is motivated from the studies of human movement and behavioral psychology [15,27], where different types of reach actions of the human have been identified and analyzed. Figure 7, shows taxonomy of such reach involving simple arm-shoulder extension (arm-and-shoulder reach), leaning forward (arm-and-torso reach) and standing reach.



Effort to Reach	Effort Level	Effort to See
No_Effort_Required	Minimum:0	No_Effort_Required
Arm_Effort		Head_Effort
Arm_Torso_Effort		Head_Torso_Effort
Whole_Body_Effort		Whole_Body_Effort
Displacement_Effort		Displacement_Effort
No_Possible_Known_Effort	Maximum:5	No_Possible_Known_Effort

Fig. 6 Effort types for visuo-spatial abilities

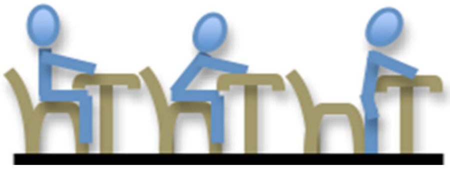


Fig. 7 Effort based taxonomy of reach action

We define a mapping operator E_t to assign a type of effort level T_E (from table of Fig. 6) to a virtual action A_v from the state S (currently represented as configuration, position and orientation) of the agent, as:

$$E_t(S, A_v) \rightarrow T_E \tag{9}$$

In the current implementation, Eq. 8 always returns the actions requiring least effort. This could be achieved in different ways: (a) By using approximate but online estimation of all the abilities by applying all the subsets of A_V from Eq. 1 as presented in [47], and then finding the least effort state among them. If the ability is satisfying from the current state itself, then A_v will be set as *No_Action_Required*. (b) Another way to find the least effort is by using an IK (Inverse Kinematics) solver iteratively by only activating the set of joints at each iteration in a predefined order corresponding to lowest to highest effort. In the current implementation we are using approach (a) as it is online and accuracy is acceptable to infer the symbolic effort related facts at the level of abstraction within the interest of the paper.

As mentioned earlier, to the best of our knowledge, such concept of multi-state visuo-spatial perspective taking, i.e. by combining effort, ability and perspective taking, has not been exploited in task understanding from demonstration.

5.1.6 Effort Based Comparative Facts: Relative Effort Class

The robot should be able to relatively analyze two efforts. For this we define an operator, which compares two effort levels and assign a class C_{RE}, as:

$$C_{RE}\left(T_E^1, T_E^2\right) = \begin{cases} Remains\_Same & if\ T_E^1 = T_E^2 \\ Becomes\_Easier & if\ T_E^1 < T_E^2 \\ Becomes\_Difficult & if\ T_E^1 > T_E^2 \end{cases} \tag{10}$$

Note that $C_{RE}\left(T_E^1, T_E^2\right) \neq C_{RE}\left(T_E^2, T_E^1\right)$

Although not used in current implementation of learning, we further have a measure of amount of effort for a particular effort level in terms of how much the agent has to turn/lean, etc. Hence, the robot could further compare two efforts of same effort level. This could be further enhanced based on the studies of musculoskeletal kinematics and dynamics models [32,56]. Whether the input is effort level or amount of effort, the robot extracts the comparative facts of Eq. 10.

5.1.7 Effort Based Qualitative Facts: Nature of Relative Effort Class

We have further enhanced the robot's knowledge-base with another layer of abstraction by qualifying the Relative Effort Classes (C$_{RE}$) as *supportive* and *not supportive*. Based on the intuitive reasoning that if an object becomes difficult to be reached by a person, the intention/nature behind it is not to support the person's ability to reach the object. Hence, we qualify the intention behind the change in effort level for an ability type *Ab*, by assigning a nature, N_{REC}^{Ab} as:

$$N_{REC}^{Ab}\left(C_{RE}^{Ab}\right) = \left\{\begin{array}{ll} S: Supportive & if\ C_{RE}^{Ab} \in \left\{\begin{array}{c} Remanins\_Same, \\ Becomes\_Easy \end{array}\right\} \\ NS: Not\_Supportive & if\ C_{RE}^{Ab} \in \{Becomes\_Difficult\} \end{array}\right. \quad (11)$$

Figure 8 shows the hierarchy of facts based on efforts.

5.1.8 Quantitative Visibility Based Hierarchy of Facts

The robot compares two visibility scores, ViS^1 and ViS^2 to obtain relative visibility score classes as:

$$C_{RViS}\left(ViS^1, ViS^2\right) = \left\{\begin{array}{ll} Almost\_Same & if (ViS^1 - ViS^2 \approx 0) \\ Increased & if\ ViS^1 << ViS^2 \\ Decreased & if\ ViS^1 >> ViS^2 \end{array}\right. \quad (12)$$

Again, we qualify the nature, N_{RViSC}, of this relative class based on whether the quantitative visibility of the object is supported or not:

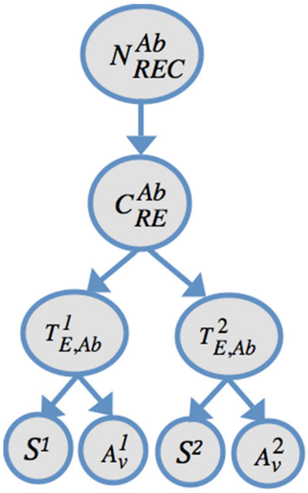


Fig. 8 Effort based Hierarchy of facts

$$N_{RViSC}\left(C_{RViS}\right) = \left\{\begin{array}{ll} S: Supportive & if\ C_{RViS} \in \left\{\begin{array}{c} Almost\_Same, \\ Increased \end{array}\right\} \\ NS: Not\_Supportive & if\ C_{RViS} \in \{Decreased\} \end{array}\right. \quad (13)$$

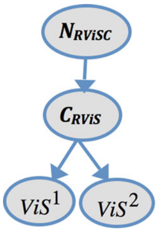


Fig. 9 Visibility scores based hierarchy of facts

Figure 9 shows the hierarchy of facts by analyzing the visibility scores.

5.2 Agent Status Based Hierarchy of Facts

5.2.1 Agent Posture Based

The robot tracks the body of the human and online distinguishes between standing and sitting postures, based on relative positions and orientations of the body parts. Agent's posture predicate *Post* is:

$$Post \in \{Standing, Sitting\} \quad (14)$$

Further, by comparing two postures, a class is assigned as:

$$C_{RPost}\left(Post^1, Post^2\right) = \left\{\begin{array}{l} M: Maintained\ \ if\ Post^1 = Post^2 \\ C: Changed\ \ otherwise \end{array}\right. \quad (15)$$

5.2.2 Human Hand Status Based

From the human's perspective a status is assigned to his/her hand as:

$$H_S \in \{Holding\_Object : OH, Free\_of\_object : OF, Resting\_on\_Support : RS\} \quad (16)$$

The robot further compares two instances of human hand status from the point of view of manipulability of the object. Based on the reasoning that if the object is in the hand, then the human can directly manipulate it, a comparative class is assigned as follows (*Manip* stands for Manipulability, see expression 16 for other abbreviations):

$$C_{RHS}\left(H_S^1 \rightarrow H_S^2\right) = \begin{cases} M: Manip\_Maintained & if\ H_S^1 = H_S^2 \wedge H_S^2 = OH \\ G: Manip\_Gained & if\ H_S^1 \neq H_S^2 \wedge H_S^2 = OH \\ L: Manip\_Lost & if\ H_S^1 \neq H_S^2 \wedge H_S^1 = OH \\ V: Manip\_Avoided & if\ H_S^1 \neq OH \wedge H_S^2 \neq OH \end{cases} \tag{17}$$

Further, a qualifying nature to a relative class, $c = C_{RHS}\left(H_S^1 \rightarrow H_S^2\right)$, from the agent's perspective is assigned as (see expression 17 for abbreviations):

$$N_{RHSC}(c) = \begin{cases} MD: Manip\_Desired & if\ c\ \in \{M, G\} \\ MND: Manip\_Not\_Desired & if\ c\ \in \{L, V\} \end{cases} \tag{18}$$

This again results into a hierarchy of facts based on human's hand status. Note that in the current implementation, if the state of either hand changes, it is treated as change in manipulability.

Since, we chose to assign only two possible values to this comparative attribute, we further do not qualify them by adding another level.

5.3 Object Status Based Hierarchy of Facts

5.3.1 Object Placement Status Based

Based on relative positions of an object with respect to the human's hand and other objects, a symbolic placement status to the object is assigned. Currently, the object placement status predicate can have the following values:

$$O_s \in \{Inside\_Container, On\_Support, In\_Hand, In\_Air\} \tag{19}$$

Any ambiguity in object placement status is resolved based on simple case based rules. Such as, if the hand is in contact with an object but the object is also on a support, it returns object status as *On_Support*.

By comparing two ordered instances of O_s a class is assigned as:

$$C_{ROS}\left(O_S^1 \rightarrow O_S^2\right) = \begin{cases} M: Maintaining\left(O_S^1\right) & if\ O_S^1 = O_S^2 \\ G: Gaining\left(O_S^2\right) \wedge L: Losing\left(O_S^1\right) & otherwise \end{cases} \tag{20}$$

Note the second case results into two simultaneous facts to encode the transition: gaining and losing states by the object. For example, for the lift task if initially the object was on support and now it is in hand, then the expression (20) will result into two facts: *Losing On_Support* state and *Gaining In_Hand* state, to encode the transition.

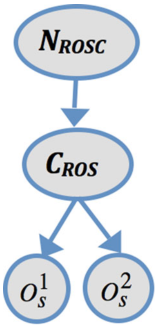


Fig. 10 Object state based hierarchy of facts

Further, we qualify the nature of the change, $c = C_{ROS}\left(O_S^1 \rightarrow O_S^2\right)$, as *supportive* to a state $O_S^{'}$ if the transition *maintains* or *gains* that state, as:

$$N_{ROSC}(c) = \begin{cases} S: Supportive\left(O_S^{'}\right) & if\ c \in \left\{M\left(O_S^{'}\right), G\left(O_S^{'}\right)\right\} \\ NS: Not\_Supportive & if\ c \in \left\{L\left(O_S^{'}\right)\right\} \end{cases} \tag{21}$$

Hence, a hierarchy of facts based on object placement states is built, as shown in Fig. 10.

5.3.2 Object Motion Status Based

As already illustrated in Fig. 1, the environment observation and inference is continuous in time. Hence, based on the temporal reasoning about the object's position, at any point of time the motion status of the object is knows as:

$$O_{ms} \in \{Moving: Mv, Static: St\} \tag{22}$$

Further, by comparing two instances of motion status, a relative status class for the object's motion state transition is assigned as follows [see expression (22) for abbreviations]:

$$C_{ROMS}\left(O_{ms}^1 \rightarrow O_{ms}^2\right) = \begin{cases} motion\_gained & if\ O_{ms}^1 = St \wedge O_{ms}^2 = Mv \\ motion\_lost & if\ O_{ms}^1 = Mv \wedge O_{ms}^2 = St \\ motion\_maintained & if\ O_{ms}^1 = Mv \wedge O_{ms}^2 = O_{ms}^1 \\ motion\_avoided & if\ O_{ms}^1 = St \wedge O_{ms}^2 = O_{ms}^1 \end{cases} \tag{23}$$

As pointed in the beginning of the section, one contribution of this paper is to identify the key aspects of reasoning about ability, effort and perspective taking based predicates (which we think are more relevant for the tasks within the scope of the paper). However, it could further be enriched by other types of facts, e.g. predicates, which can capture the effects of type: *putting on the top of a particular object or at a particular place.*

In this section we have enriched the robot's knowledge-base with a set of hierarchy of facts related to the human and the object, which is inferred by the robot online. Next section will describe our generalized task-understanding framework based on *explanation-based learning* and *m-estimate based* *refinement*. The framework takes into account such hierarchies of facts and autonomously learns the tasks' semantics at appropriate levels of abstractions.

6 Explanation Based Task Understanding

In addition to have human-level understanding of the task and to make such understanding independent of how the task has been executed, another motivation behind the current work is to enable the robot to begin learning even from the single positive demonstration and refine with subsequent demonstrations. Therefore, we have adapted the framework of *Explanation Based Learning (EBL)* (see the survey [65]), which has been shown to possess the desired characteristics and could be used for concept refinement (i.e. specialization) as well as concept generalization [19]. For continuity, below we mention the components of a typical *EBL* system (see [19] for detail):

Goal Concept: A definition of the concept to be learned. Given in terms of high-level properties, which are not directly available in the representation of an example.

Training Example: A lower level representation of the examples.

Domain Theory: A set of inference rules and facts sufficient for providing that a training example meets the high-level definition of the concept.

Operationality Criterion: Defines the form in which the learned concept definition must be expressed.

Generally domain theory and operationality criterion are devised to restrict the allowable learned vocabulary and initial hypothesis space, to ensure that the new concept is 'meaningful' to the problem solver (the task planner).

Our approach is similar to *EBL* [23,65], in the following manner:

It (1) constructs an explanation tree for each example of a task, (2) compares these trees to find largest sub tree, and (3) forms a horn clause using the leaf nodes of the largest sub tree to find the general rule.

However, towards addressing the issue of scalability, our approach will differ from *EBL* in the sense, instead of providing a proper domain theory and operationality criterion for the target-concept to precise the hypothesis space, we will provide a general *goal concept* in terms of the effect of the task. This will create a hypothesis space with wide ranging knowledgebase of the robot, spanning over lowest to highest levels of abstractions. This will ensure to learn any task, which could possibly incorporate any of the effect related predicates known to the robot. Then based on the demonstrations, the robot has to autonomously refine/prune the hypothesis space. This will prevent providing separate domain theory for each and every task the robot will encounter, as well as will enable the robot to autonomously extract relevant features for a particular task.

We are following the framework of the explanation based learning, which requires providing a goal concept and a domain theory (a set of inference rules and facts sufficient for providing that a training example meets the high-level definition of the concept). That is why we have to provide a hypothesis space in terms of the knowledgebase, which encodes the domain theory in terms of various inference rules. In the future work, we will address the aspect of generating a hypothesis space based on different knowledge gathering approaches, such as [36,60].

6.1 General Target Goal Concept to Learn

We provide for any task *tk*, performed by a performing-agent P_{ag} for a target-agent T_{ag} on a target-object T_{obj}, the generalized goal concept to learn as:

$$Task\,(tk) \leftarrow \mathit{effect}\,(WI, WF) \tag{24}$$

As illustrated in Fig. 1 *WI* and *WF* are snapshots of the continuously inferred facts and continuously observed world states at time stamps t_i and t_f marking the start and the end of a demonstration.

6.2 Provided Domain Theory

Further, the following *domain theory* is provided:

$$\begin{aligned} \mathit{effect}\,(WI, WF) \leftarrow N^{reach}_{REC}\left(T_{ag}, T_{obj}\right) \wedge \\ N^{grasp}_{REC}\left(T_{ag}, T_{obj}\right) \wedge N^{see}_{REC}\left(T_{ag}, T_{obj}\right) \\ \wedge N_{RVis}\left(T_{obj}, T_{ag}\right) \wedge C_{RPost}\left(T_{ag}\right) \wedge \\ N_{RHSC}\left(T_{ag}\right) \wedge N_{ROSC}\left(T_{obj}\right) \wedge C_{ROMS}\left(T_{obj}\right) \end{aligned} \tag{25}$$

The task is learned in the form of *desired effects* from 'any' *target-agent's* perspective for 'any' *target-object*.

The above expression when mapped into the definitions of inferred facts discussed earlier in this paper, results into the following representation:

$$\begin{aligned} \mathit{effect}\,(WI, WF) \leftarrow \\ \mathit{Nature\_Effect\_Class\_to\_Reach}\left(T_{ag}, T_{obj}\right) \wedge \\ \mathit{Nature\_Effect\_Class\_to\_Grasp}\left(T_{ag}, T_{obj}\right) \wedge \\ \mathit{Nature\_Effect\_Class\_to\_See}\left(T_{ag}, T_{obj}\right) \wedge \\ \mathit{Nature\_Visibility\_Score}\left(T_{obj}, T_{ag}\right) \wedge \\ \mathit{Effect\_Relative\_Posture}\left(T_{ag}\right) \wedge \\ \mathit{Nature\_Effect\_Hand\_Status}\left(T_{ag}\right) \wedge \\ \mathit{Nature\_Effect\_Object\_Status}\left(T_{obj}\right) \wedge \\ \mathit{Effect\_Object\_Motion\_Status}\left(T_{obj}\right) \end{aligned} \tag{26}$$

The definitions of the *domain theory* are already presented in expressions of Sect. 5.

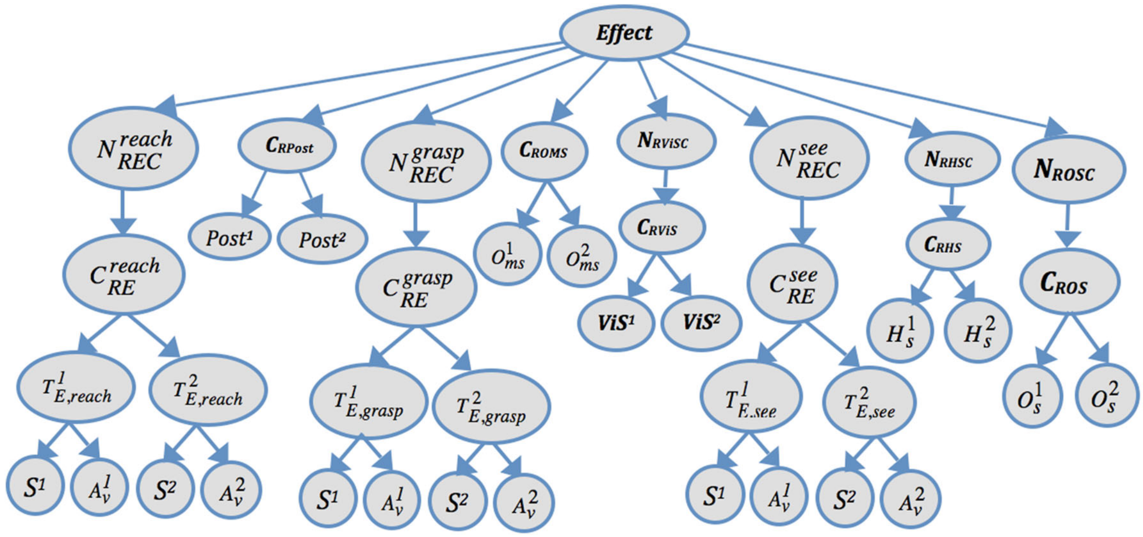


Fig. 11 Initial generalized hypothesis space for human-level effect-based understanding of tasks' semantics. The symbolic mapping of the notations from the *top–down* order from *left to right*: N_{REC} : *Nature of Relative Effort Class**, C_{RE} : *Class of Relative Effort**, T_E : *Type of Effort**, S : *State**, A : *Action**, (* : *common for the first, third and sixth subtrees for different abilities of reach, grasp and see)* C_{RPost} = *Class of Relative Posture*, $Post$: *Posture*, C_{ROms} = *Class of Relative Object motion state*, O_{ms} = *Object motion state*, N_{RViSC} = *Nature of Relative Visibility Score Class*, C_{RViS} = *Class of Relative Visibility Score*, ViS = *Visibility Score*, N_{RHsC} = *Nature of Relative Hand status Class*, C_{RHs} = *Class of Relative Hand status*, H_s = *Hand status*, N_{ROsC} = *Nature of Relative Object status Class*, C_{ROs} = *Class of Relative Object status*, O_s = *Object status*

Above domain theory when unfolded results into a *general initial hypothesis space* as shown in Fig. 11.

The *training examples* are provided in the form of lowest level representation, i.e. in 3D world model consisting of the positions and configurations of the objects and the agents. As the robot continuously observes and infers the environment, based on the time stamps of the start and the end of a demonstration, the robot autonomously instantiates the hierarchies of facts within the domain theory.

Further, to be generalized enough to learn different tasks; we do not strictly provide the form of the learned concept as *operationality criterion*. It could be composed of any set of nodes of the initial hypothesis space as shown in Fig. 11.

6.3 m-Estimate Based Refinement

Each node of initial hypothesis space of Fig. 11 serves as a predicate. For refining the learned concept based on multiple demonstrations, instead of directly pruning the explanation sub-tree based on getting two different values for a node, we use *m-estimate* based reasoning. *m-estimate* has been shown to be useful for rule evaluation [26] and to avoid premature conclusions [1]. This is because the generalized definition of *m-estimate* incorporates the notion of *experience*, as described below.

Let us say a value v for a particular predicate p for a particular task tk has been observed in n number of demonstrations, out of a total of N demonstrations. The likelihood (i.e. the measure of the extent to which a sample provides support for particular values) of observing the same value v for the next demonstration within the *m-estimate* framework will be given as:

$$Q_p^{v,tk}\,(n, N) = \frac{n + a}{N + a + b} \tag{27}$$

where, $a > 0, b > 0, a + b = m \; and \; a = m \times p_v$

m is domain dependent, and could also be used to include noise [12]. From Eq. 27 following properties could be deduced:

$$Q_p^{v,tk}\,(0, 0) = P_v > 0 \tag{28}$$

$$Q_p^{v,tk}\,(0, N) = \frac{a}{N + a + b} > 0 \tag{29}$$

$$Q_p^{v,tk}\,(N, N) = \frac{N + a}{N + a + b} < 1 \tag{30}$$

Hence, it does not assume a close world in the sense if it did not observe a value v for predicate p, it does not mean that likelihood of the existence of v is NULL [expressions (28), (29)]. In fact, p_v is prior probability of v, as also can be inferred from Eq. 28. On the other hand, if always the same value v has been observed, that too will not be accepted as universal rule that p will always have the value v for the task tk [expression (30)]. Hence, it takes into account the likelihood of unseen demonstrations. These properties allow lifelong refinement of the learned concept.

$$Q_p^{v,tk}\,(N + 1, N + 1) > Q_p^{v,tk}\,(N, N) \tag{31}$$

Above property [expression (31)] ensures that even the value *v* has been observed for all the examples, the likelihood to observe same value will be more if more number of examples has been demonstrated, thus incorporating the notion of *experience*.

$$Q_P^{v,tk}\,(0, N) < Q_P^{v,tk}\,(0, N+1) \tag{32}$$

This property ensures that even the value *v* has never been observed, the likelihood that *v* will *not* be observed in the future will be less if less number of examples have been demonstrated, thus again incorporating the notion of *experience*.

One acceptable instantiation of *m-estimate* is using *Laplace's law of succession*. This states that if in the sample of *N* trials, there were *n* successes, the likelihood of the next trial being successful is *(n + 1)/(N + 2)*, assuming that the initial distributions of success and failure are uniform. With the similar initial assumption, we also use *a = 1* and *a + b = 2* for *m-estimate* of Eq. 27.

6.4 Consistency Factor

As the robot is required to autonomously reason about whether a predicate *p* is relevant or not, it analyzes the consistency in the observed value of the predicate. If the values are not always the same, it means the predicate might not be relevant for that task and the values are just the side effects, not the desired effect. We further assume that v_h is the value for *p* having the highest *m-estimate* obtained from Eq. 27. If this value is consistent over demonstrations, then the predicate *p* is relevant and its desired value will be v_h. Let, for a particular predicate *p*, over *N* demonstrations, N_p different values $\{v1, v2, v3, \ldots v_{Np}\}$ have been observed. We define a *consistency factor (CF)* of *p* for task *tk* to decide about the relevance of *p* as:

$$CF_p^{tk} = \overbrace{Q_p^{v_h,tk}}^{\text{relevanceevidence}} - \underbrace{\sum_{i=1\wedge i\neq h}^{Np} Q_p^{v_i,tk}}_{\text{non}-\text{relevanceevidence}} \tag{33}$$

The first part on the right side of the equation shows the evidence of *p* being relevant for the task. Higher this value, greater will be the likelihood that the most observed single value, v_h, for *p* is the part of the desired effect for task *tk*. The second part gives the likelihood of obtaining any of the observed value other than v_h. This in fact represents *non-relevant evidence* of *p*, NRE_p, because, higher this value, lower the likelihood of *p* having a consistent value. Hence, based on the value of the consistency factor after any demonstration, we define following three situations about a particular predicate *p* for a particular task *tk* (see Fig. 12):

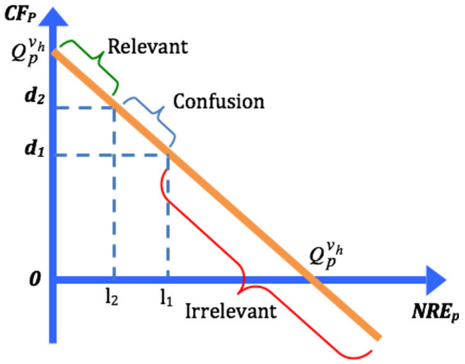


Fig. 12 Deciding relevance and irrelevance of a predicate, as well as potential confusion

(i) *Contradiction, so irrelevant p*: A predicate *p* will be assumed to be non-relevant based on contradiction in its value, (a) if $CF < 0$; non-relevant evidences are collectively higher than the relevant evidence, or (b) if $0 <= CF <= d_1$; non-relevant evidences are significant to contradict the likelihood of v_h being the expected consistent value of *p*.

(ii) *Consistency, so relevant p*: if $CF > d_2$; as the non-relevant evidences are significantly lower and could be ignored.

(iii) *Confusion, require clarification*: if $d_1 < CF < d_2$; as the non-relevant evidences are not sufficient to contradict the current understanding but also not low enough to be ignored directly. In this case, the robot has to ask the human partner for clarification about the significance of the predicate *p* and its desired value by framing a sentence including the values causing the confusions.

It is evident that by setting different values on the threshold and maintaining separate boundaries allow to tune d_1 and d_2 based on various practical factors such as the reliability of the demonstration, the accuracy of the inferred fact, noise at different levels of the system, nature, sensitivity and criticality of the domain and attributes, preferences on inconsistency tolerances, etc.

However, as the demonstrations are assumed to be positive, which means we will not try to teach a child with wrong examples, little evidence of a predicate assuming different values should be sufficient to prune that node from the explanation tree. This assumption results into almost coinciding d_1, d_2 and $Q_p^{v_h}$. Therefore, we have set d_2 based on the *10 %* tolerance of the inconsistency in relevant predicate, i.e. $l_2 = 0.1 \times Q_p^{v_h}$, hence:

$$d_2 = Q_p^{v_h} - 0.1 \times Q_p^{v_h} \tag{34}$$

Similarly, we have given the robot greater autonomy to decide a predicate to be irrelevant if there exists non-relevant evidence as low as *30 %* of the relevant evidence, i.e.

$l_1 = 0.3 \times Q_p^{v_h}$ which results into:

$$d_1 = Q_p^{v_h} - 0.3 \times Q_p^{v_h} \tag{35}$$

Setting the above-mentioned very narrow band of confusion zone is mainly for two purposes: (1) Demonstrations are assumed to be positive and observation of the world is assumed to be almost noise free at the symbolic level. (2) Our robot was not equipped with sufficient speech synthesis and dialogue module and no feedback interface was installed in front of the demonstrators. Therefore, we wanted to avoid any explicit communication, instead to let the demonstrator to focus on the task and the robot to focus on observing demonstration and inference of facts.

At this point, it is important to note that the robot keeps track of *m-estimate* of all the observed values for all the predicates to maintain the notion of inter-value experiences, even currently a particular value might have been found irrelevant. This facilitates the robot to incorporate experience and allow modifying its understandings lifelong.

7 Demonstrations and Analysis

7.1 Experimental Platform

The robot uses *Move3D* [59], an integrated 3D planning and visualization platform. Through its various sensors, the robot maintains and updates the 3D world state in real time. For object identification and localization, it uses tags based stereovision system. For localizing and tracking humans, it uses data from 3D motion sensors mounted on it. The human's gaze is simplified to his/her head orientation, estimated through markers tracked by a marker based motion capture system in real time.

We have tested our system on two different robots: *JIDO* a home-built mobile manipulator equipped with a LWR Kuka arm and *PR2* robot from Willow Garage. As shown in Fig. 13a, b, Jido and PR2 robots are observing the environment. Figure 13c, d show the 3D world representation of the environment built and updated online by the robots.

7.2 Procedure

For each demonstration, the human operator provides the name of the task, indicates the *performing-agent*, the *target-agent* and the *target-object*, and tags the time stamps of start and finish of the task.

Note that the *target-agent* is the agent the task is targeted to, not the target of the experiment. As the robot should be able to perceive the necessary and sufficient parts of the environment before and after the demonstration. Therefore, in most of the cases the placement of the robot was closer to the *performing-agent*. This perception coupled with our perspective taking system, which allows the robot to reasons from the perspective of the *target-agent*, provides sufficient information to analyze the effect of the demonstration.

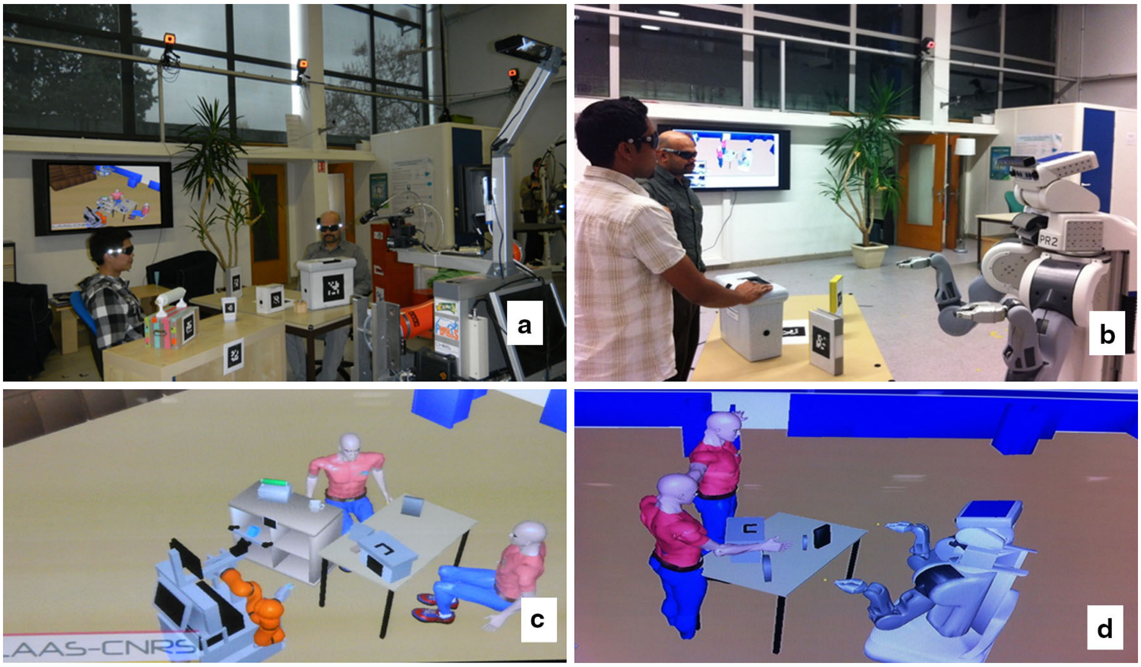


Fig. 13 Mobile robots **a** JIDO and **b** PR2 are observing Human–Human interaction scenario. **c**, **d** 3D representation of the world built and updated online by the robots

The demonstrations were the part of a bigger set of three complementary studies: The semantics understanding, which is the focus of this paper, trajectory level analysis of performing different manipulation tasks by different people, and the preferred places to perform the tasks with respect to both the agents in different setups. Although in our case, it would have been sufficient to have demonstrations by a single expert teacher, because of the other aspects the demonstrations were performed by different demonstrators. However, since we did not tell the demonstrators about where to perform the task, it adds some inherent diversity to further demonstrate the working of the presented framework.

The demonstrators were the researchers from our robotics lab selected based on their availability. They are aware about the perception system of the robot; hence, they know how to hold the object (i.e. not to hide the tag) so that the robot can track it most of the time. They have been told about the name of the task they are expected to perform and some hints about some of the confusing aspects. For example, most of them have been told about the main difference between *make accessible* and *give*, that the object should be *placed* somewhere in the former case. However, most of them did not know the algorithm and what was being learned. There were three to five demonstrations for each task and data used in this paper are taken at least from two different demonstrators for each task.

In the current approach, the name of the target-object is explicitly provided to the robot. However, works on autonomous learning of task-relevant objects such as [35] could be adapted for this purpose.

In all the demonstrations, the explanation tree has been constructed by inferring the facts from the *target-agent*'s perspective. This is to find the desired effect for the person for whom the task has to be performed. However, a similar tree could be constructed from the perspective of the *performing-agent* as well, to find how the agent prefers to perform the task.

As explained in Sect. 6, the robot constructs an explanation tree for each new demonstration of the task, by instantiating the hypothesis tree of Fig. 11. For instantiating the leaf nodes, the predicates with superscript *1* represent the data from *WI*, the initial world state, whereas superscript *2* represent the data from *WF*, the final world state.

Greater the diversity among the demonstrations for the same task, faster the non-relevant predicates will be pruned out from the task's understanding. To achieve the diversity across demonstrations we used to change the scenarios by changing the relative arrangements of the *performing-* and *target- agents*, the *target-object* and its initial position, and the positions of other objects and furniture.

Next we will discuss the resulting understanding of the robot for different tasks.

7.3 Show an Object

The first task demonstrated to the robot was to *show* an object. Figure 14a–d show final scenarios of four different demonstrations of the task. The red quadrilaterals show initial positions of the *target-object* [which is cup in (a) and wooden cube in (b), (c) and (d)], the red arrows mark the final position of the *target-object* at the end of the task. In situations (a) and (c) the *target-agent* was the person on the left, whereas, for (b), he was the *performing-agent*. In (d) the person on the right was the *target-agent*. The largest consistent sub-tree after first two demonstrations (a) and (b) has been shown in Fig. 15. Below each node of the tree the corresponding inferred values of the predicates have been shown in parenthesis {}. The learned target concept for the task is obtained in terms of horn clause from the leaf nodes of this sub-tree.

Figure 16 shows partial instantiation of the hypothesis space for the individual demonstration of Fig. 14c. And Fig. 17 shows the refined explanation, based on the largest common, *m-estimate* based consistent, sub-tree for all the three demonstrations. In the fourth demonstration for the same task, a different pair of performing and target agents demonstrated the task in standing postures. The performing-agent has put the target-object, the wooden cube, on another object, white box, to make it visible, as shown in Fig. 14d. Figure 18 shows the refined explanation tree. The refined understanding after these four demonstrations, formed by the horn clause of the leaf nodes is:

$$\begin{aligned} Task\,(Show\_Object) \leftarrow (C_{RPost} = Maintained) \wedge \\ (O^1_{ms} = Static) \wedge (O^2_{ms} = Static) \wedge \\ (C_{RVis} = Increased) \wedge \\ (A^2_v(see) = No\_Action\_Required) \wedge \\ (H^1_s = Object\_Free) \wedge (H^2_s = Object\_Free) \end{aligned} \tag{36}$$

By replacing the abbreviations with the symbolic terms, presented in Sect. 5, the above expression results into the following:

$$\begin{aligned} Task\,(Show\_Object) \leftarrow \\ (Relative\_Posture = Maintained) \wedge \\ (Object\_Initial\_Motion\_Status = Static) \wedge \\ (Object\_Final\_Motion\_Status = Static) \wedge \\ (Object\_Relative\_Visibility\_Score = Increased) \wedge \\ (Action\_to\_See = No\_Action\_Required) \wedge \\ (Initial\_Hand\_Status = Object\_Free) \wedge \\ (Final\_Hand\_Status = Object\_Free) \end{aligned} \tag{37}$$

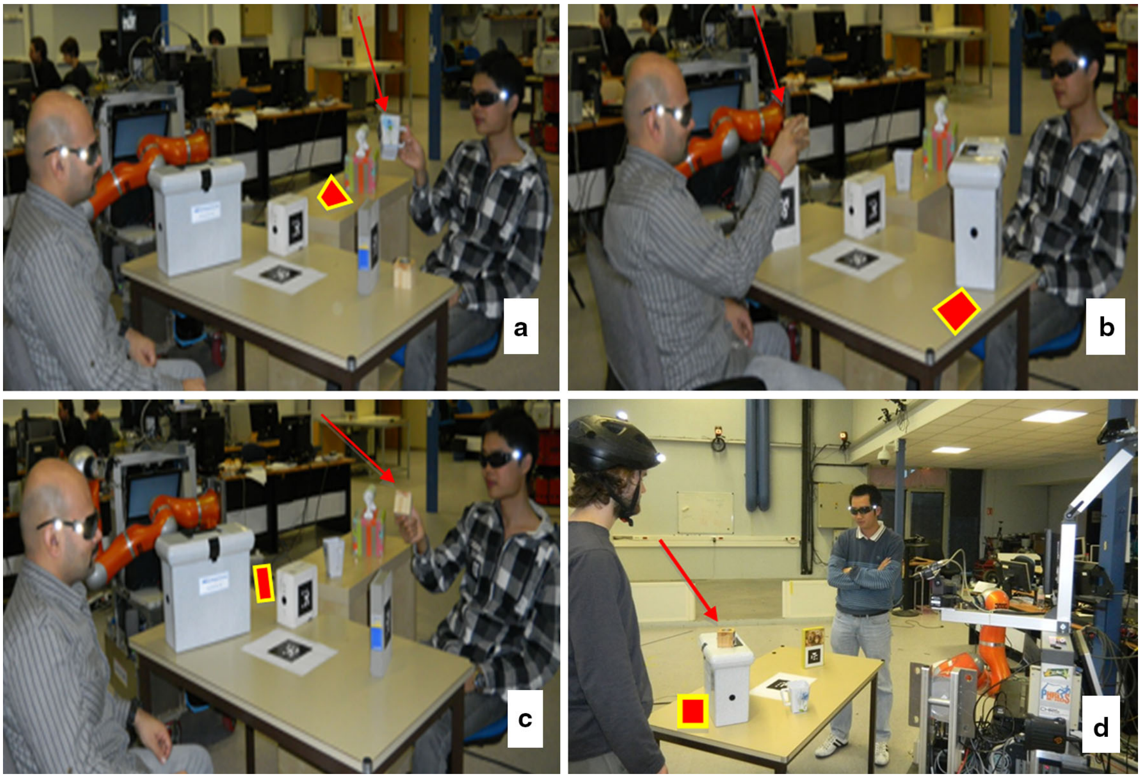


Fig. 14 Human–Human demonstration for *show* an object task. Initial positions of the *target-object* are shown by red quadrilaterals. **a** Right human is showing the cup by holding it. **b** Left human is showing the wooden cube by holding it. **c** Right human is showing the wooden cube by holding it. **d** Left human is showing the wooden cube by making it visible by putting it on the *top* of the *white box*. (Color figure online)

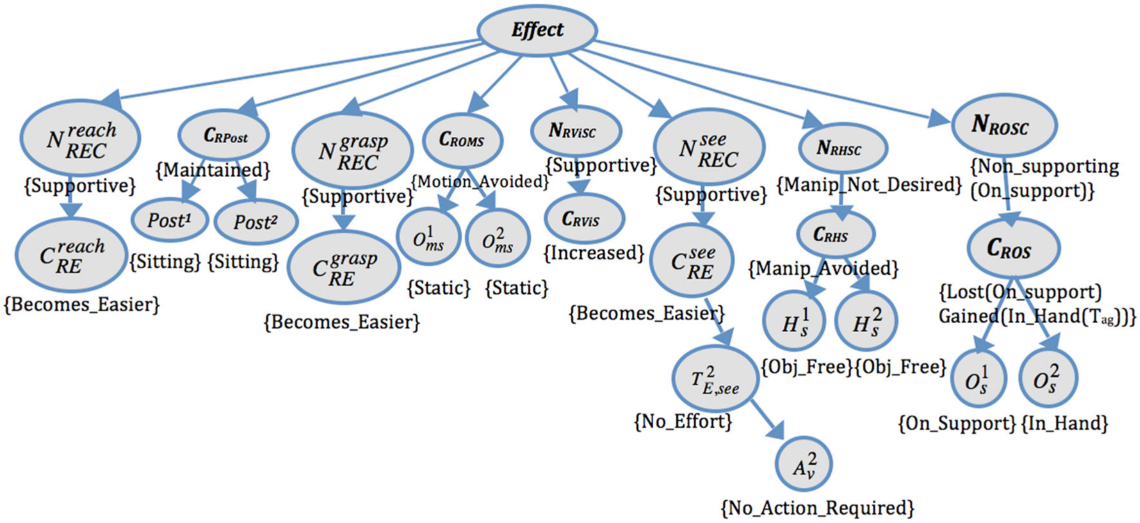


Fig. 15 Explanation tree for the *show* task after two demonstrations **a** and **b** of Fig. 14

Note that the above understanding is from the *target-agent's* perspective. This means the *target-agent* should put no effort to see the *target-object*, the visibility score of the *target-object* should be increased from the *target-agent's* perspective, the hand of the *target-agent* should be free of object, etc. Also note that the irrelevant predicates such as reachability and graspability of the *target-agent* as well as the object's status have been autonomously pruned out from the learned desired effect of the task.

Explicitly learning the preconditions for a task is not the scope of this paper, however, when a leaf node corresponding to *WI* appears in the learned concept, we let it be there to be

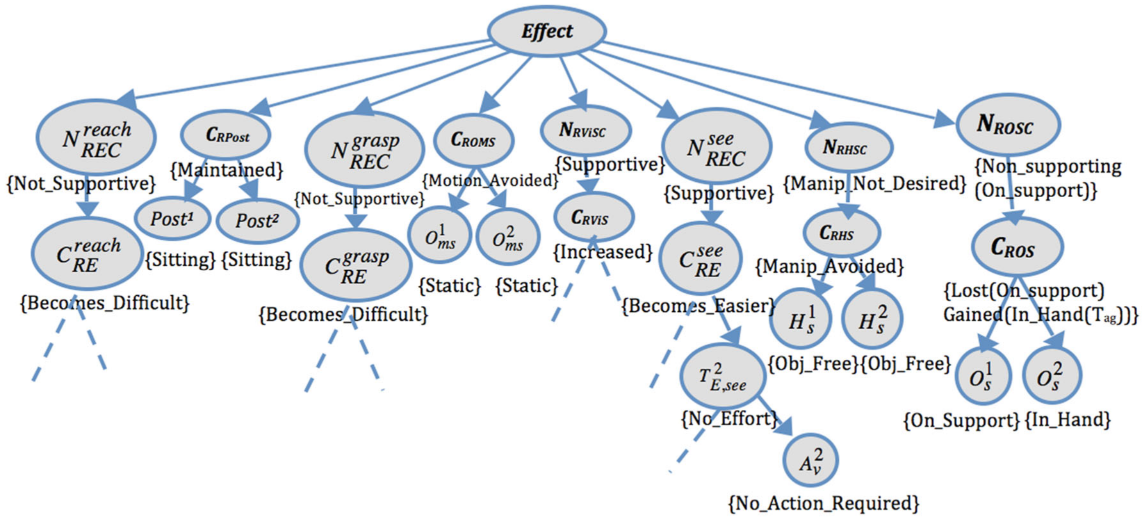


Fig. 16 Partial instantiation of the hypothesis space of Fig. 11 for explaining the *show object* task of demonstration (**c**) of Fig. 14. Note the main difference in the left most sub-trees, compared to the explanation tree of Fig. 15

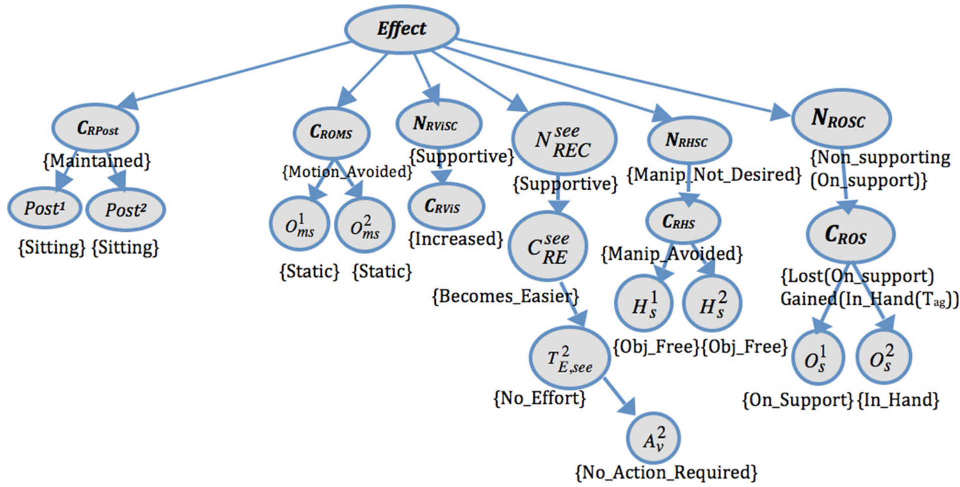


Fig. 17 Refined consistent explanation tree after the three demonstrations, **a**, **b** and **c**, of Fig. 14 for the *show object* task

useful for task planners to enrich the list of preconditions based desired effects.

Remark on m-estimate and Consistency Factor: Table 2 analyzes the consistency of node O^2_S of explanation tree, which corresponds to the final object status for the *show object* task demonstrations. Fourth column shows the *m-estimate* corresponding to the value *v* of the predicate O^2_S having highest *m-estimate*. And fifth column shows the simple probability based estimation of value *v*. By comparing both the columns we observe an interesting difference. *m-estimate* as explained earlier takes into account *experience*. Hence, with each new demonstration in which value $v = In\_Hand$ (first three demonstrations) is obtained, the robot's belief to have the same value is increasing, based on experience. Whereas, in the case of probability it is always the same as 1, hence, simple probability fails to update experience based belief. This observation illustrates the property of expression

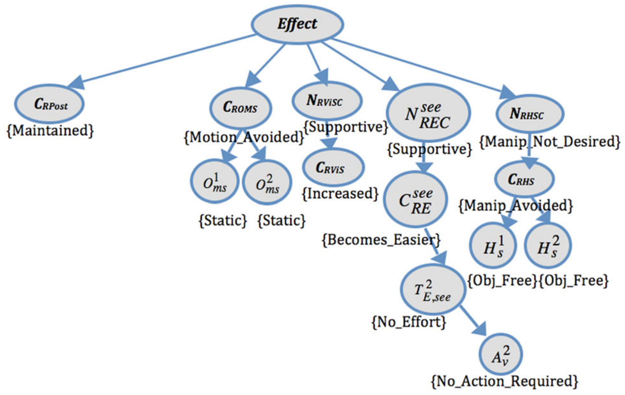


Fig. 18 Refined consistent explanation tree after the fourth demonstrations, (**d**), of Fig. 14 for the *show object* task. Note that the abstraction level for C_{RPost} has been changed, compared to the understanding of Fig. 17

Table 2 Consistency factor analysis

Consistency analysis of predicate Object State after demonstration (O_s^2) for show task

Demo	Current observed value	Value (v) corresponding to highest m-estimate	m-estimate based relevance evidence (RE) for v	Probability based value of v, P(v)	Non-relevance evidence (NRE)	Confusion boundary limit on Consistency Factor		Predicate's current consistency factor (CF)
						Min (dl)	Max (d2)	
1	ln_Hand	ln_Hand	0.66667	1	0	0.46667	0.6	**0.66667**
2	ln_Hand	ln_Hand	0.75	1	0	0.525	0.675	**0.75**
3	ln_Hand	ln_Hand	0.8	1	0	0.56	0.72	**0.8**
4	On_Support	ln_Hand	0.66667	0.75	0.33333	0.46667	0.6	**0.33334**

Bold values correspond to the main data to observe, and is evident in the text

(31) of *m-estimate*, which states that: even the same value has been observed in all the demonstrations, the expectation to observe the same value in next demonstration will be increasing with more number of demonstrations.

Further, our approach does not fix the value of the thresholds across demonstrations. This can be observed by analyzing the columns seven and eight. Thresholds *d1* and *d2* of Fig. 12 are dynamically varying. This is because of our approach, which decides these thresholds based on the *m-estimate* of the relevance evidence, Eqs. (34) and (35). Note that in the last column, for the first three demonstrations, the consistency factor is well above *d2*, so the robot considers the predicate and its value as relevant. Now analyze the row corresponding to the fourth demonstration. This time a different value has been observed for the predicate, which is *On_Support*, but still the value corresponding to the highest *m-estimate* is *In_Hand*. After fourth demonstrations, the consistency factor of highest m-estimate value of the predicate is *0.33334*. This is well below *d1*, the minimum threshold for confusion and makes the robot sure about its irrelevance.

Note that we did not observe any confusion because of our deliberately selected very narrow *confusion zone*. As we have already mentioned we assume the demonstrations to be positive, so a little inconsistency makes the predicate to be irrelevant. However, if we decide to give less decisional autonomy to the robot, forcing the robot to ask for clarification instead of directly deciding the irrelevance, we could make the *confusion zone* wider. For example, if we set the range to be between 10 and 60 % of the relevance evidence instead of 10–30 %, in that case, the same demonstration 4 will result into a confusion, as *d1* will become *0.26667* hence, consistency factor *0.33334* will lie between *d2* and *d1*.

7.4 Hide an Object

The next task demonstrated to the robot was *hide an object*. Fig. 19 shows three demonstrations for this task with different initial scenarios. The understanding of the task, after these three demonstrations, formed by the horn clause of the leaf nodes, results into the following symbolic representation:

$$
\begin{aligned}
&Task\,(Hide\_Object) \leftarrow \\
&(Human\_Initial\_Posture = Sitting) \wedge \\
&(Human\_Final\_Posture = Sitting) \wedge \\
&(Object\_Initial\_Motion\_Status = Static) \wedge \\
&(Object\_Final\_Motion\_Status = Static) \wedge \\
&(Object\_Final\_Visibility\_Score \approx 0) \wedge \\
&(Relative\_Effort\_Class\_to\_See = Becomes\_Difficult) \wedge \\
&(Initial\_Hand\_Status = Object\_Free) \wedge \\
&(Final\_Hand\_Status = Object\_Free) \wedge \\
&(Object\_Initial\_Status = On\_Support) \wedge \\
&(Object\_Final\_Status = On\_Support)
\end{aligned}
\tag{38}
$$

Similar to the *show* task, further demonstrations in which the *target-agent* would be standing, would result into refined understanding related to the *target-agent*'s posture attribute. However, note that the main differences between the understandings of the *show* and the *hide* tasks have been captured. In the *hide* task the effort hierarchy corresponding to the ability to see the object is pruned at relative effort class level, instead of maintaining the lowest level node, of required action, as was the case of *show* task. This results into the understanding of the *hide* task that the relative effort to see the object should become difficult for the *target-agent*. Also the visibility score based subtree has been pruned at lowest level of absolute value of visibility score. For the *show* task, the actual values of visibility score from the *target-agent*'s perspective were not consistent but were always greater than the initial values. Hence, the framework autonomously results into a higher level of abstraction for the *show* task, which is: increased relative visibility score. Whereas, in the case of the *hide* task the absolute value of visibility score itself is

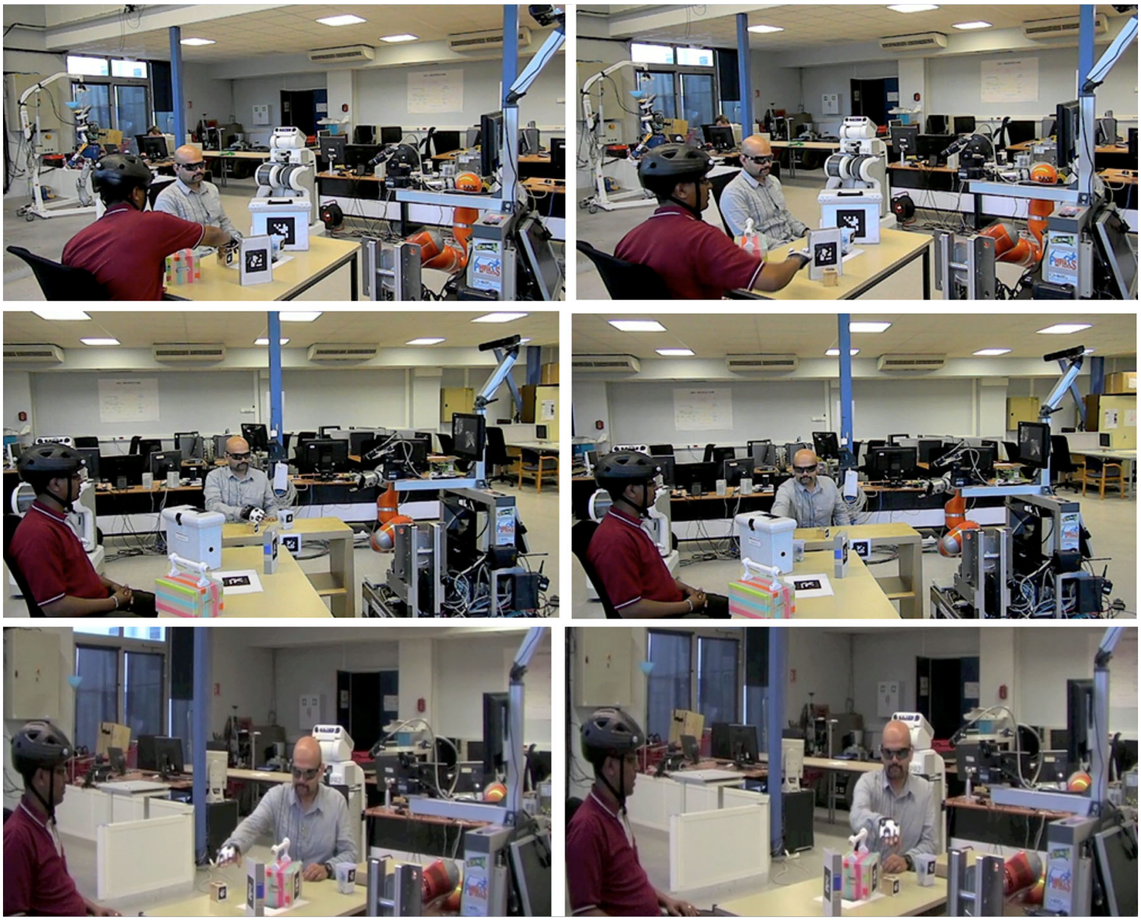


Fig. 19 Three demonstrations for the task of hiding an object, observed by JIDO robot. *First column* shows the initial scenarios and the *second column* shows the final scenarios after performing the task

always negligible, hence, making the visibility score node as consistent and relevant.

Note that again the effect on the abilities to reach and grasp the object by the *target-agent* have been autonomously found to be irrelevant and pruned out from the explanation tree, as was the case for the *show* task.

7.5 Make an Object Accessible

Next, the task of *making an object accessible* has been demonstrated to the robot. There were a total of five demonstrations, three of them were similar to the demonstrations of Fig. 2b–d. The rest two demonstrations were in different relative arrangements of the objects and the humans, and the *target-agent* was in standing posture.

As the intention behind *make-accessible* task is to provide the *target-agent* with the flexibility to take the object sometime in future whenever required. Therefore, the provided end time stamps of the *make-accessible* demonstrations were the instants when the *performing-agent* finishes the task by putting the object on the table to make it accessible, not the instants shown in Fig. 2b–d, where the *target-agent* has already begun to reach the *target-object*.

The understanding of the robot about the *make-accessible* task after these five demonstrations results into the following symbolic representation:

$$\begin{aligned}
Task\,(Make\_Accessible) \leftarrow &\\
(Relative\_Effort\_to\_Reach = Becomes\_Easier) \wedge &\\
(Relative\_Posture = Maintained) \wedge &\\
(Object\_Initial\_Motion\_Status = Static) \wedge &\\
(Object\_Final\_Motion\_Status = Static) \wedge &\\
(Relative\_Effort\_to\_Grasp = Becomes\_Easier) \wedge &\\
(Object\_Relative\_Visibility\_Score = Increased) \wedge &\\
(Nature\_Effort\_Class\_to\_See = Supportive) \wedge &\\
(Initial\_Hand\_Status = Object\_Free) \wedge &\\
(Final\_Hand\_Status = Object\_Free) \wedge &\\
(Object\_Initial\_Status = On\_Support) \wedge &\\
(Object\_Final\_Status = On\_Support) &
\end{aligned} \tag{39}$$

Note that the predicate related to the posture of agent is from the point of view of desired effect on the posture due to the task. This posture does not indicate the change of posture, which might occur due to the actions required by the *target-agent* to see, reach or grasp the *target-object*. This is captured in other set of attributes encoded in effort-based hierarchy, for example $Relative\_Effort\_to\_Reach$. The *Relative_Posture* predicate obtained in the above understanding indicates that the task does not change the posture of the agent.

The interesting observation for the *make-accessible* task understanding is that, it did not filter out reachability and graspability as irrelevant predicates, as were the cases for the *show* and *hide* tasks. It understands that the reachability and graspability of the *target-object* by the *target-agent* should become *easier*.

7.6 Give an Object

The next task demonstrated to the robot was to *give* an object. The scenarios were similar to the *make-accessible* task, the only difference was that the *performing-agent* was holding the *target-object* at appropriate place in the space and waiting for the *target-agent* to take it, instead of putting the object on the support. For this task the end time stamps indicated to the robot were the moments when the *target-agent* takes the object from the *performing-agent*. There were a total of three demonstrations and the task understanding based on the leaf nodes of the *m-estimate* based consistent explanation subtree results into the following symbolic representation:

$$\begin{aligned}
&Task\ (Give) \leftarrow \\
&(Action\_to\_Reach = No\_Action\_Required) \wedge \\
&(Relative\_Posture = Maintained) \wedge \\
&(Action\_to\_Grasp = No\_Action\_Required) \wedge \\
&(Object\_Initial\_Motion\_Status = Static) \wedge \\
&(Object\_Final\_Motion\_Status = Static) \wedge \\
&(Object\_Relative\_Visibility\_Score = Increased) \wedge \\
&(Nature\_Effort\_Class\_to\_See = Supportive) \wedge \\
&(Relative\_Hand\_Status = Manipulability\_Gained) \wedge \\
&(Object\_Initial\_Status = On\_Support) \wedge \\
&(Object\_Final\_Status = In\_Hand)
\end{aligned} \tag{40}$$

Compared to the *make-accessible* task, the main differences (which in-fact are interrelated) in the understanding of the *give* task are: the *target-agent* should apply no action to reach and grasp the *target-object*, the *target-object* should be in the hand of *target-agent* and the manipulability of the *target-object* should be gained by the *target-agent*. Further, it encodes that the *give* task will not be finished until the *target-object* is in *target-agent*'s hand, whereas for the *make-accessible* task it is sufficient to make the *target-object* easier to be reached and grasped by the *target-agent*.

7.7 Put-Away an Object

Next we have demonstrated the task of *put-away* an object. There were four demonstrations in different situations. Following is the robot's understanding about the *put-away* task:

$$\begin{aligned}
&Task(Put\_Away) \leftarrow \\
&(Relative\_Effort\_to\_Reach = Becomes\_Difficult) \wedge \\
&(Relative\_Posture = Maintained) \wedge \\
&(Object\_Initial\_Motion\_Status = Static) \wedge \\
&(Object\_Final\_Motion\_Status = Static) \wedge \\
&(Relative\_Effort\_to\_Grasp = Becomes\_Difficult) \wedge \\
&(Relative\_Visibility\_Score = Decreased) \wedge \\
&(Relative\_Effort\_to\_See = Maintained) \wedge \\
&(Initial\_Hand\_Status = Object\_Free) \wedge \\
&(Final\_Hand\_Status = Object\_Free) \wedge \\
&(Object\_Initial\_Status = On\_Support) \wedge \\
&(Object\_Final\_Status = On\_Support)
\end{aligned} \tag{41}$$

7.8 Hide-Away an Object

Next task to demonstrate was to *hide-away* an object. Again there were four demonstrations in different situations, resulting into the following understanding:

$$\begin{aligned}
&Task\ (Hide\_Away) \leftarrow \\
&(Relative\_Effort\_to\_Reach = Becomes\_Difficult) \wedge \\
&(Relative\_posture = Maintained) \wedge \\
&(Object\_Initial\_Motion\_Status = Static) \wedge \\
&(Object\_Final\_Motion\_Status = Static) \wedge \\
&(Relative\_Effort\_to\_Grasp = Becomes\_Difficult) \wedge \\
&(Object\_Final\_Visibility\_Score \approx 0.0) \wedge \\
&(Relative\_Effort\_to\_See = Becomes\_Difficult) \wedge \\
&(Initial\_Hand\_Status = Object\_Free) \wedge \\
&(Final\_Hand\_Status = Object\_Free) \wedge \\
&(Object\_Initial\_Status = On\_Support) \wedge \\
&(Object\_Final\_Status = On\_Support)
\end{aligned} \tag{42}$$

Note that the above understanding of the *hide-away* task tries to inherit the properties of the *hide* and *put-away* tasks. For the *hide* task the abilities to reach and grasp were found to be irrelevant, whereas in the *hide-away* task, similar to the *put-away* task, these have been found relevant in the sense to make these abilities difficult for the *target-object* from *target-agent*'s perspective. For the *put-away* task, the robot found relative visibility score to be decreasing because of the relatively away position of the *target-object*, whereas for *hide-away* task the absolute visibility score itself has been found to be negligible from the *target-agent*'s perspective, as was the case for the *hide* task. Hence, the framework prunes the subtrees at different appropriate levels of abstractions.

In these resulted understandings of the tasks, the effects related to the abilities to reach and to grasp, wherever appeared in the learned concepts, were similar. This is because of the type of task demonstrated. However, if the robot will be demonstrated with the tasks such as, put the *target-object* to enable the *target-agent* just to touch it, the framework would successfully capture the independent and different effects about the abilities to reach and to grasp.

8 Performance Analysis

In this section, we will analyze the performance from the perspective of the average processing time and based on intuitive understanding of the tasks. Here, it is important to note that as mentioned earlier, the numbers of demonstrations mentioned in the following figures are representing the demonstrations by expert teachers, that too by manually achieving diversity among different demonstrations for the same task, for faster convergence. Following our analogy that we are not teaching a 'child' with wrong demonstration, analysis of experiments with non-expert teacher is not the scope of the paper.

8.1 Processing Time

Table 3 shows number of demonstrations per task and the average processing time per demonstration (excluding the time of demonstration). It is interesting to observe that it takes more time for those tasks, which require the robot to apply more number of virtual actions on the *target-agent* for finding the least feasible effort for an ability. For example, in the case of the *hide* task, the *target-object* had been placed to be invisible by the *performing-agent* from the perspective of *target-agent*. In most of such cases, the robot needs to apply *Whole_Body_Effort* or even *Displacement_Effort* to find the least feasible effort to see or reach the *target-object* by the *target-agent*, after sequentially testing for lower effort levels. Whereas, for the tasks for which the least feasible effort of the *target-agent* is found by applying the virtual actions corresponding to lower-level efforts such as *Head_Effort or Arm_Effort*, the computation time is less. For example, for the *give* task, from *target-agent*'s perspective, least efforts to see and reach are both lower, hence, lower processing time.

Table 3 Number of demonstration per task and average learning/refinement time (in *s*) after each demonstration for each task

Task	Total number of demonstrations	Average time for learning after each demonstration in *s*
Make accessible	5	0.56
Hide	3	0.81
Show	4	0.63
Give	3	0.59
Put away	4	0.61
Hide away	4	1.2

In fact, the convergence after each demonstration depends upon the total number of predicates in the domain theory, as even the learned concept appear to be pruned significantly, the robot maintains the *m-estimate* of all the predicates to incorporate the possibility of lifelong learning and confusion based refinement. This is a choice we have made. However, one could chose to refine only the tree learned so far and batch process the remaining data offline. This will make the system learn faster but needs to decide *what* to batch process and *when*.

8.2 Analyzing Intuitive and Learned Understanding

In this paper, we have tried to incorporate a subset of predicates for better human-level understanding of a task, but we cannot claim it to be a complete domain theory for understanding of such tasks. We do not follow a close world assumption and in fact we should not, as a task could have effects, which cannot be captured in the current hypothesis space, and could include other physical, mental and emotional states of the *target-agent*, his/her desire and so on. For example, does making accessible require the *target-agent* to maintain his/her posture or his/her foot to be on ground, should the *target-object* always be visible more easily or is it acceptable to compromise it for the ease of reachability. Therefore, it is practically not possible to figure out an exact ground truth model, hence, to define a domain theory, which will be "complete" or "accurate".

The question of "correct tree" will probably be unanswered until we will have a "complete" domain knowledge and universally accepted "precise semantics" of tasks. Hence, it is difficult to design an objective function for analysis and comparison. Therefore, in this paper we can only intuitively say something about a "partially correct" explanation of a task for comparison purpose.

Since, the focus of the paper is on understanding tasks, which are performed for other agents, in this section we have chosen the predicates related to the effect on *target-agent's* ability and the effect on *target-object*'s state to analyze the "partial correctness". For this, based on an intuitive understanding of tasks, we have compared the robot's

Table 4 Analyzing the key attributes' understanding by the robot with an intuitive understanding for a task

Task	Intuitive understanding of task			Number of demonstration to reach at intuitive understanding			Number of trial to get the intuitive sub-tree of hypothesis
	For target agent's ability		For target object state	For target agent's ability		For target object state	
	To see	To reach		To see	To reach		
Make Accessible	Supportive	Easier	Static on support	4	2	1	4
Hide	Difficult	(Not relevant)	Static on support	2	2	1	2
Show	Directly	(Not relevant)	(Not relevant)	1	3	4	4
Give	Supportive	No action*	In target-agent's hand	3	1	1	3
Put Away	Maintain	Difficult	Static on support	2	2	1	2
Hide Away	Difficult	Difficult	Static on support	3	3	1	3

* Assuming task ends when object is grasped by target agent

understanding. The intention is to demonstrate the strength of the presented framework that it can reach such human-level understanding even from two positive demonstrations, if demonstrated as differently as feasible. Table 4 summarizes this comparison. Note that for reaching the understanding at higher level abstraction such as *supportive*, it takes more number of trials than the lower level predicates based understanding, such as *directly*, which maps to *No_Effort*, a value corresponding to lower level in the effort hierarchy. This is obvious, as the pruning of the sub-trees are bottom up for the sake of understanding a task at appropriate level of abstraction and generalization. Therefore, generally the number of demonstrations to conclude non-relevance accumulates as the level of abstraction goes up.

It is worth to note that even though as explained earlier, it is difficult to have an objective function and correct tree to compare, the intuitive understanding used in this section and the analysis of the result from different perspectives as done in the columns of Tables 3 and 4 could serve to analyze and compare the results of future works on understanding such tasks.

8.3 Necessity of Ability Based Facts Based and Perspective Taking

Let us assume that the hypothesis tree does not contain any attribute related to the ability based on the perspective taking of the *target-agent*. In that case, the understood semantics for the tasks of *Make Accessible* and *Put Away* are:

$$\begin{aligned} &Task\,(Make\_Accessible) \leftarrow \\ &(Relative\_Posture = Maintained) \wedge \\ &(Object\_Initial\_Motion\_Status = Static) \wedge \\ &(Object\_Final\_Motion\_Status = Static) \wedge \\ &(Initial\_Hand\_Status = Object\_Free) \wedge \\ &(Final\_Hand\_Status = Object\_Free) \wedge \\ &(Object\_Initial\_Status = On\_Support) \wedge \\ &(Object\_Final\_Status = On\_Support) \end{aligned} \tag{43}$$

$$\begin{aligned} &Task(Put\_Away) \leftarrow \\ &(Relative\_Posture = Maintained) \wedge \\ &(Object\_Initial\_Motion\_Status = Static) \wedge \\ &(Object\_Final\_Motion\_Status = Static) \wedge \\ &(Initial\_Hand\_Status = Object\_Free) \wedge \\ &(Final\_Hand\_Status = Object\_Free) \wedge \\ &(Object\_Initial\_Status = On\_Support) \wedge \\ &(Object\_Final\_Status = On\_Support) \end{aligned} \tag{44}$$

Hence, these two tasks, which are opposite in nature are not semantically separated, because the core difference is from the perspective of the *target-agent* on the ability to reach the object. Therefore, the reasoning about abilities from the *target-agent*'s perspective is a must for understanding those types of tasks in which an agent is performing for another agent.

8.4 Importance of Levels of Abstraction

Let us take the example of the learned semantics of the tasks *hide* and *show*. In the show task, the sub-tree related to the ability to see the *target-object* has been preserved at the lowest level of *target-agent*'s required action (*Action_to_See* = *No_Action_Required*). This results into an intuitive notion that the *target-agent* should not put any effort to see the object. Whereas, for the *hide* task, the same sub-tree has been pruned to relative effort class level (*Relative_Effort_Class_to_See* = *Becomes_Difficult*), which results into an understanding that the *target-object* should become difficult to be seen by the *target-agent*, and the effort to see can vary depending upon the situation. Similarly, for the task of *make accessible* the same sub-tree has been pruned at the highest level, i.e. the nature of the effect, (*Nature_Effort_Class_to_See* = *Supportive*), which can also be defended as initially the object might be hidden or already visible and the *making accessible* might increase the visibility or at least maintain it, while reducing the effort to be reached by the *target-agent*. Hence, even there is inconsistency in the values at relative effort level in the hierarchy, the nature/intention has been captured at the higher level of abstraction.

In the above example if we do not have the leaves at action level, the difference between the *show* and the *hide* tasks would have been still captured at the relative effort class to see (i.e. easy or difficult to see). However, the finer notion that the *target-agent* should put no effort to see the object in the case of *show* task would be lost. Similarly, if we do not have the qualifying the level of abstraction, *nature*, the inconsistency at the level of relative effort to see would result into an understanding, which will consider the ability to see the object as an irrelevant predicate. Hence, the design of the knowledgebase with significant levels of abstractions is important to capture the intuitive semantics of the tasks, to avoid missing some relevant predicates as well as to avoid losing some finer details.

8.5 Remark on Practical Limitations

The practical issues related to inference of various facts limit the output of the presented framework. For example, the *object_in_hand* fact is inferred if the object is not on a support and the hand is close to the object. This limits estimating how the object is being grasped by the agent, i.e. what is the relative position and orientation of the hand with respect to the grasped object. This further limits the understanding about dual grasp, i.e. there should be sufficient space available to grasp the object simultaneously by another human. Therefore, for the *give* task we have deliberately chosen the object of bigger dimensions. This always leaves sufficient space, so that the robot can positively infer a collision free grasp by the *target-agent*. Another limitation arises from the localization of the object. Since it is based on the tag on the object, the *performing-agents* were instructed to manipulate the object such that the tag always faces towards the camera of the robot.

9 Potential Applications and Benefits

The symbolic understanding of a task along with its geometric counterpart makes the robot more 'aware' about a task. In this section we discuss some of the potential applications.

9.1 Generalization to Novel Scenario

The understanding of a task is independent of the shape and size of the object, the trajectory as well as the absolute/relative distances among the agents and objects. This facilitates the robot to perform the task in an entirely different scenario. For example, the robot will be able to perform the task of making an object accessible by putting it at the top of another object, even it might have not been demonstrated to the robot.

9.2 Greater Flexibility for Symbolic and Shared Cooperative Plan Generation

If the planner at symbolic level knows the semantics of a task independent of the actions to be executed, it could plan to achieve the task in a variety of ways. Such as, if it 'understands' that hiding means the *target-object* should be difficult to be seen by the *target-agent*, depending upon the situation, it could plan to cover the *target-object* with some other container type object to make it invisible. Similarly, for showing or making an object accessible, again instead of directly manipulating the *target-object*, it could plan to displace the occluding or obstructing object from the human's perspective to achieve the same desired effects. The task-planner could even involve a third agent to achieve the task.

Goal effect based task planners, such as [2,3,8,17], and affordance based task planners [5,48], could take into account high-level constraints and generate alternative solution in terms of actions, agents and objects for a task. Our ongoing work in this direction, such as [17,18,48], demonstrates the feasibility of a hybrid planner capable of generating cooperative shared plans for achieving a task in human–robot interaction domain, based on combined geometric and symbolic reasoning. It is able to take into account the aspects such as which agent can do which type of action, what are the associated levels of efforts, constraints and preferences. Nevertheless, it will be an interesting future work to incorporate the complementary aspects of action learning from demonstrations, and incorporate that through a mechanism to reduce the complexity or at least guide the search space of such task planner.

9.3 Transfer of Understanding Among Heterogeneous Agents

Since the robot understands the task independent of the trajectory planning and control level execution, it can easily transfer the task semantics to another robot of entirely different kinematics structure and shape. And the other robot equipped with similar reasoning capabilities could then interpret the understanding and perform it by respecting its own constraint of whole body planning.

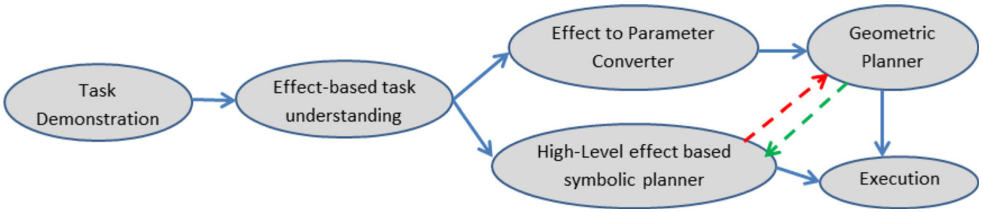

Task Demonstration
Effect-based task understanding
Effect to Parameter Converter
Geometric Planner
High-Level effect based symbolic planner
Execution

Fig. 20 Different ways to reproduce a learned task, based on desired effect. Either the effect could be converted in terms of constraints for a geometric pick-and-place task planner or could be used by high-level symbolic planner to produce entirely different plans depending upon the situation

9.4 Feasibility of Reproducing the Learned Task

As the learning is not at trajectory or sub-action level, so it is not possible to reproduce the task by imitation approaches. In fact, the domain of effect-based task planning is a complementary research topic.

As shown in Fig. 20 a task learned in terms of desired effect could be reproduced in different ways:

(i) *Effect-to-Parameter Converter*: By using a converter, which will interpret the effect and convert it into the parameters of a geometric task planner.
(ii) *High-Level task Planner*: By using a High-Level task planner, such as HATP [2] and other as discussed in Sect. 9.2 to find the actions, which could result into the desired learned effect the world. The advantage of using such high level symbolic planner is, it can plan to achieve same effect in different ways, as discussed in Sect. 9.2. As shown in dotted link, a two-way handshaking between the low-level planner and the high-level planners, such as [3,17] could better converge to a feasible plan, based on desired effect and effort.

In [47] we have presented a framework to find the candidate places to perform a task, once the semantics of the task is known in terms of effort based ability to see and reach. In [49], we have presented a framework to plan for such basic HCOM tasks. In [51] we presented an integrated planner based on constraint hierarchy, for planning a set of basic pick-and-place tasks and presented experimental results. The core of the planner takes as input three sets: candidate places, candidate grasps, candidate orientation of the object. Further, it takes the range of visibility score, need of dual grasp as input parameter, to test for feasibility of a solution.

As the domain of the demonstrated tasks of the current paper is within the scope of our planner presented in [51], we have implemented a simple converter to ground the learned effects in terms of the parameters of this planner. This converter has different sub-modules to convert different predicates. For example, to convert the symbolic value “easier” for the ability “reach” of the *target-agent*, it finds the *target-agent*’s current least effort, C_{LE} to reach the *target-object*. Then it finds the places, which are reachable by less effort levels than C_{LE} by the *target-agent*, using approach in [47]. These places serve as the input set of candidate places. Similarly, the value of desired visibility score is directly passed. If the object status should be *on_support*, this is also passed directly, as the framework is capable of finding set of stable placement on any support plane. As the effect of task understanding is at a level of abstraction, which makes it independent from the kinematics of the agent, we have taken this advantage to feed the converted constraints for planning for the *hide* task by PR2 robot, although learned by JIDO robot. Fig. 21 shows PR2 robot reproduces the task of hiding a *target-object* (grey-tape) form a *target-agent* (human) in a different scenario.

Note that the planner of [51] only serves as an example, which could be used in geometric planner block of Fig. 20. For a different geometric planner, an appropriate effect to parameter converter will be required.

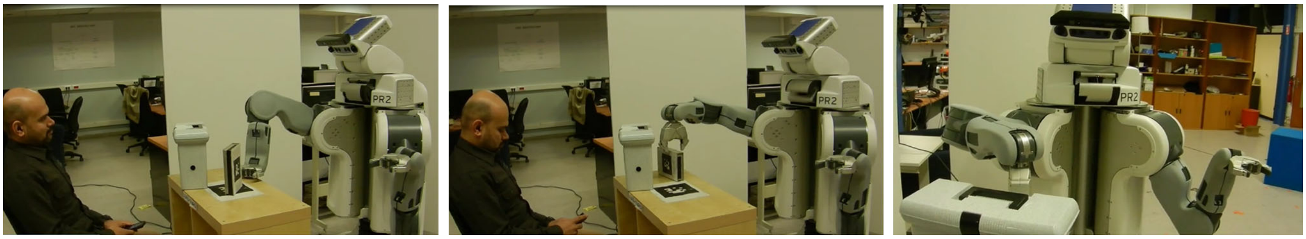


Fig. 21 Using the effect to parameter converter, learned effects of the *hide* task has been converted into constraints comparable with our geometric planner for human-centered tasks [51]. Using those, the PR2 robot reproduces the task of hiding a *target-object* (Grey tape) from a *target-agent* (Human) in a different scenario

9.5 Understanding by Observing Heterogeneous Agents

Our system is capable of effort based visuo-spatial perspective takings analysis not only for human but also for different robots, *JIDO*, *HRP2*, *PR2*. Therefore, using the same framework and system, the robot could understand task semantics from the demonstrations in which the *target-agent* is a robot. This facilitates understanding for the situation in which the human might perform the task for the observer robot itself or for another type of robot.

9.6 Generalization for Multiple Target-Agents

Another interesting future research work is to incorporate the aspect of multiple *target-agents*. Such as, hide an object from two humans at the same time, show an object to a group of people, etc. There will be two different aspects:

(i) Adapting the understanding to perform for multiple *target-agents*: The symbolic level understanding of tasks will facilitate easy generalization for planning for multiple *target-agents*, but will depend upon the capability of the task planner to plan and perform the task for multiple *target-agents*.

(ii) Understanding from the demonstrations involving multiple *target-agents*: From the point of view of understanding the task, which itself has been aimed for multiple *target-agents*, the framework can be adapted to build and maintain multiple hypotheses trees for different *target-agents*. The underlying research challenge will be merging such hypotheses tress to autonomously come up with a coherent understanding.

9.7 Generalization for Multiple Target-Objects

In the current implementation, we focused on the tasks requiring a single *target-object* of interest. There could mainly two ways in which multiple *target-objects* could be important for a task:

(i) Without any relation or ordering: For example, giving something to write, which required giving pen and paper. For this our current framework can be adapted, so that the robot will reason on the effects for different target-objects, independent of each other. This might also lead to autonomous reasoning about identifying the *target-objects*. In the current approach, the name of the *target-object* is explicitly provided to the robot. Works on autonomous learning on task-relevant objects such as [35] could be adapted for this purpose.

(ii) There are some semantics associated with the relation and ordering between the objects: For example, serving coffee, which will require to put the cup above the plate. For this, it will require integrating the presented approach with the works on learning ordering and preferences, such as [52].

9.8 Facilitate Task/Action Recognition and Proactive Behavior

By partially observing the human's action and its effect, the robot could probabilistically classify the task or the desired changes in the ongoing action, by the human. Even the complete task is known to the robot, again based on the desired effects of the task, the robot could show proactive behaviors to partially/fully facilitate the task while reducing the human's effort. For example, if the robot infers/knows that the human is trying to reach an object, it could proactively offer help by making the object accessible. Similarly, if the robot knows at symbolic level that the human wants to show or give an object to it, knowing the 'meaning' of task, the robot could proactively turn its head or move its arm to facilitate achieving the desired effect as an attempt to guide as well as support the task [50].

9.9 Enriching Human–Robot Natural Interaction

Such symbolic awareness about the task's semantics could also enrich the interaction with the human partner to be more natural, as the robot will be able to communicate the task at the level of abstraction understandable by the human.

9.10 Understanding Other Types of Tasks

The focus of the paper is on basic HCOM tasks, which requires one agent to perform the task for the other agent. We do not claim that with its current knowledgebase the system can learn every manipulation task. However, the presented knowledge and the learning framework can help in augmenting the knowledgebase for effect-based understanding of various other types of tasks, such as *tap*, *lift*, *drop*, *dump* in trash bin, *throw* an object, etc. For example after tapping a ball initially laying *static* on a support, the final state of the ball is *moving* on support, hence, the effect contains object symbolic status: $on\_support$, motion status: *moving*. Similarly, for *throw* task the effect contains object symbolic status: in_air, motion status: *moving*. Similarly, for *lift* task, the object status $on\_support$ will be lost and object status in_hand will be gained, hence, manipulability gained, etc. Certainly more predicates and reasoning about the dynamic effects will be required to understand the complete effect of such tasks, such as *dump*, which requires the notion of "place" or "bin". However, if the domain theory and the knowledgebase are rich enough (either engineered or learnt by some ontological representation) the presented framework could be used to autonomously prune out irrelevant facts, associated either with the human or with the object.

10 Conclusion and Future Work

This paper serves as a step towards making the boundary between task primitives and execution primitives visible and enables a human-centered robot to understand task semantics independent of the means to achieve it. This is an important aspect of *emulation* learning, which facilitates us, the humans, to efficiently perform a task differently in different situations.

In this paper we focused on those tasks, which require one agent to perform a task for another agent, e.g. *show*, *hide*, *give*, etc. The main novelty of the paper is that we have argued and demonstrated that it is a must to include *multi-state visuo-spatial perspective taking*, i.e. combining the reasoning about effort, ability and perspective taking for understanding such tasks by a robot. We have identified the key attributes, which enable the robot to reason on various quantitative, comparative and qualitative facts for analyzing the effect of demonstrated tasks. An *explanation based learning* (EBL) framework, incorporating *m-estimate* based consistency measure, which introduces the notion of 'experience', has been presented for understanding task semantics from demonstration. All these equip the robot with the ability to learn and explain its understanding in a 'meaningful' way. We have demonstrated that the robot autonomously learns the intuitive semantics of a set of basic human-centered tasks at appropriate levels of abstractions and argued that such understanding could be generalized to novel scenarios and heterogeneous agents.

A work in progress is to integrate the learning framework with goal-effect based task planners to reproduce the task in an entirely different way. Further, the presented approach could benefit from various other attributes to understand more complex tasks.

The presented framework currently reasons at initial and final world state $\langle WI, WF\rangle$ to understand a task based on its effect. An interesting future work is to include the action component A and reason based on $\langle WI, A, WF\rangle$. This will enable the robot to analyze the rich information about execution preferences and to learn those tasks for which some sub-action primitives might be essential parts for understanding the desired effect of the task. Further, it will be interesting to investigate, how the framework could be adapted to understand the *undesirable effects* of a task.

References

1. Agostini A, Torras C, Wörgötter F (2011) Integrating task planning and interactive learning for robots to work in human environments. In: Proceedings of the 22nd international conference on artificial intelligence (IJCAI), pp 2386–2391
2. Alili S, Alami R, Montreuil V (2008) A task planner for an autonomous social robot. In: Distributed autonomous robotic systems (DARS), pp 335–344
3. Alili S, Pandey A, Sisbot EA, Alami R (2010) Interleaving symbolic and geometric reasoning for a robotic assistant. In: ICAPS workshop on combining action and motion planning (CAMP)
4. Argall BD, Chernova S, Veloso M, Browning B (2009) A survey of robot learning from demonstration. Robot Auton Syst 57(5):469–483
5. Awaad I, Kraetzschmar GK, Hertzberg J (2013) Affordance-based reasoning in robot task planning. In: Planning and robotics (PlanRob) workshop at 23rd international conference on automated planning and scheduling (ICAPS)
6. Breazeal C, Berlin M, Brooks AG, Gray J, Thomaz AL (2006) Using perspective taking to learn from ambiguous demonstrations. Robot Auton Syst 54:385–393
7. Calinon S, Dhalluin F, Caldwell D, Billard A (2009) Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework. In: 9th IEEE-RAS international conference on humanoid robots, humanoids 2009, pp 582–588
8. Cambon S, Alami R, Gravot F (2009) A hybrid approach to intricate motion, manipulation and task planning. Int J Robot Res 28(1):104–126
9. Cantrell R, Schermerhorn P, Scheutz M (2011) Learning actions from human-robot dialogues. In: IEEE RO-MAN, pp 125–130
10. Carello C, Grosofsky A, Reichel FD, Solomon HY, Turvey M (1989) Visually perceiving what is reachable. Ecol Psychol 1(1):27–54
11. Carpenter M, Call J (2002) The chemistry of social learning. Dev Sci 5(1):22–24
12. Cestnik B (1990) Estimating probabilities: a crucial task in machine learning. In: Proceedings of the ninth European conference on artificial intelligence, ECAI, pp 147–149
13. Chao C, Cakmak M, Thomaz A (2011) Towards grounding concepts for transfer in goal learning from demonstration. In: IEEE international conference on development and learning (ICDL) vol 2, pp 1–6
14. Chella A, Dindo H, Infantino I (2006) A cognitive framework for imitation learning. Robot Auton Syst 54(5):403–408
15. Choi HJ, Mark LS (2004) Scaling affordances for human reach actions. Hum Mov Sci 23(6):785–806
16. Dantam N, Essa I, Stilman M (2012) Linguistic transfer of human assembly tasks to robots. In: Proceedings of intelligent robots and systems (IROS), pp 237–242
17. de Silva L, Pandey AK, Alami R (2013) An interface for interleaved symbolic-geometric planning and backtracking. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 232–239
18. De Silva L, Gharbi M, Pandey AK, Alami R (2014) A new approach to combined symbolic-geometric backtracking in the context of human–robot interaction. In: IEEE international conference on robotics and automation (ICRA)
19. Dejong G, Mooney R (1986) Explanation-based learning: an alternative view. Mach Learn 1:145–176
20. Dillmann R (2004) Teaching and learning of robot tasks via observation of human performance. Robot Auton Syst 47:109–116
21. Dragan A, Gordon G, Srinivasa S (2011) Learning from experience in manipulation planning: setting the right goals. In: Proceedings of the international symposium on robotics research (ISRR)
22. Ekvall S, Kragic D (2008) Robot learning from demonstration: a task-level planning approach. Int J Adv Robot Syst 5(3):223–234
23. Flann NS, Dietterich GT (1989) A study of explanation-based methods for inductive learning. Mach Learn 4:187–226

24. Flavell JH, Everett BA, Croft K, Flavell ER (1981) Young children's knowledge about visual perception: further evidence for the level 1–level 2 distinction, pp 99–103
25. Flavell JH, Shipstead SG, Croft K (1978) Young children's knowledge about visual perception: Hiding objects from others. Child Dev 49(4):1208–1211
26. Furnkranz J, Flach P (2003) An analysis of rule evaluation metrics. In: Proceedings of the 20th international conference on machine learning (ICML). AAAI Press, pp 202–209
27. Gardner DL, Mark LS, Ward JA, Edkins H (2001) How do task characteristics affect the transitions between seated and standing reaches? Ecol Psychol 13(4):245–274
28. Gribovskaya E, Khansari-Zadeh S, Billard A (2011) Learning nonlinear multivariate dynamics of motion in robotic manipulators. Int J Robot Res 30(1):80–117
29. Hopper LM, Lambeth SP, Schapiro SJ, Whiten A (2008) Observational learning in chimpanzees and children studied through ghost-conditions. Proc R Soc B Biol Sci 275(1636):835–840
30. Johnson M, Demiris Y (2005) Perceptual perspective taking and action recognition. Int J Soc Robot 2:181–199
31. Jkel R, Schmidt-Rohr S, Rhl S, Kasper A, Xue Z, Dillmann R (2012) Learning of planning models for dexterous manipulation based on human demonstrations. Int J Soc Robot 4(4):437–448
32. Khatib O, Demircan E, Sapio VD, Sentis L, Besier T, Delp S (2009) Robotics-based synthesis of human motion. J Physiol Paris 103:211–219
33. Kuniyoshi Y, Inaba M, Inoue H (1994) Learning by watching: extracting reusable task knowledge from visual observation of human performance. IEEE Trans Robot Autom 10(6):799–822
34. Lallee S, Lemaignan S, Lenz A, Melhuish C, Natale L, Skachek S, van Der Zant T, Warneken F, Dominey PF (2010) Towards a platform-independent cooperative human–robot interaction system: I. perception. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 4444–4451
35. Lee KH, Lee J, Thomaz AL, Bobick AF (2009) Effective robot task learning by focusing on task-relevant objects. In: Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems. IEEE Press, Piscataway, NJ, USA, pp 2551–2556
36. Lemaignan S, Ros R, Mösenlechner L, Alami R, Beetz M (2010) Oro, a knowledge management module for cognitive architectures in robotics. In: Proceedings of IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 3548–3553
37. Lempers JD, Flavell ER, Flavell JH (1977) The development in very young children of tacit knowledge concerning visual perception. Genet Psychol Monogr 95(1):3–53
38. Levas A, Selfridge M (1984) A user-friendly high-level robot teaching system. In: IEEE international conference on robotics and automation. proceedings, vol 1, pp 413–416
39. Lopes M, Melo FS, Montesano L (2007) Affordance-based imitation learning in robots. In: IROS, pp 1015–1021
40. Lunsky LL (1965) Learning and instinct in animals. Arch Intern Med 115(6):757–758
41. Marin-Urias L, Sisbot E, Pandey A, Tadakuma R, Alami R (2009) Towards shared attention through geometric reasoning for human robot interaction. In: 9th IEEE-RAS international conference on humanoid robots, humanoids 2009, pp 331–336
42. Michael L (2011) Causal learnability. In: Barcelona, International joint conference on artificial intelligence (IJCAI), pp 1014–1020
43. Montesano L, Lopes M, Bernardino A, Santos-Victor J (2007) Modeling affordances using bayesian networks. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 4102–4107
44. Muhlig M, Gienger M, Hellbach S, Steil JJ, Goerick C (2009) Task-level imitation learning using variance-based movement optimization. In: IEEE international conference on robotics and automation, pp 1177–1184
45. Nielsen M (2006) Copying actions and copying outcomes: social learning through the second year. Dev Psychol 42(3):555
46. Ogawara K, Takamatsu J, Kimura H, Ikeuchi K (2003) Extraction of essential interactions through multiple observations of human demonstrations. IEEE Trans Ind Electron 50(4):667–675
47. Pandey AK, Alami R (2010) Mightability maps: a perceptual level decisional framework for co-operative and competitive human-robot interaction. In: International conference on intelligent robots and systems (IROS) year 2010, pp 5842–5848
48. Pandey AK, Alami R (2013) Affordance graph: a framework to encode perspective taking and effort based affordances for day-to-day human–robot interaction. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 2180–2187
49. Pandey AK, Alami R (2014) Ingredients and a framework of dexterous manipulation skills for robots in human centered environment and hri. J Robot Soc Jpn 32(4):31–37
50. Pandey AK, Ali M, Alami R (2013) Towards a task-aware proactive sociable robot based on multi-state perspective-taking. Int J Soc Robot 5(2):215–236
51. Pandey AK, Saut JP, Sidobre D, Alami R (2012) Towards planning human-robot interactive manipulation tasks: task dependent and human oriented autonomous selection of grasp and placement. In: Proceedings of IEEE RAS & EMBS BioRob, pp 1371–1376
52. Pardowitz M, Dillmann R (2007) Towards life-long learning in household robots: the piagetian approach. In: IEEE 6th international conference on development and learning (ICDL), pp 88–93
53. Pasula H, Zettlemoyer L, Kaelbling L (2004) Learning probabilistic relational planning rules. In: International conference on automated planning and scheduling (ICAPS), pp 73–82
54. Piaget J (1945) Play, dreams, and imitation in childhood. Norton, New York
55. Ros R, Lemaignan S, Sisbot EA, Alami R, Steinwender J, Hamann K, Warneken F (2010) Which one? grounding the referent based on efficient human–robot interaction. In: 19th IEEE international symposium in robot and human interactive communication, pp 570–575
56. Sapio V, Warren J, Khatib O (2006) Predicting reaching postures using a kinematically constrained shoulder model. In: Lennarcic J, Roth B (eds) Advances in robot kinematics. Springer, Berlin, pp 209–218
57. Saut JP, Sidobre D (2012) Efficient models for grasp planning with a multi-fingered hand. Robot Auton Syst 60(3):347–357
58. Schmidt-Rohr S, Losch M, Dillmann R (2010) Learning flexible, multi-modal human–robot interaction by observing human–human–interaction. In: IEEE RO-MAN, pp 582–587
59. Simeon T, p. Laumond J, Lamiraux F (2001) Move3d: a generic platform for path planning. In: 4th international symposium on assembly and task planning, pp 25–30
60. Tenorth M, Beetz M (2009) Knowrob- knowledge processing for autonomous personal robots. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 4261–4266
61. Tomasello M (1990) Cultural transmission in the tool use and communicatory signaling of chimpanzees?. Cambridge University Press, Cambridge, MA, pp 274–311
62. Trafton J, Cassimatis N, Bugajska M, Brock D, Mintz F, Schultz A (2005) Enabling effective human–robot interaction using perspective-taking in robots. Syst Man Cybern Part A Syst Hum IEEE Trans 35(4):460–470
63. Weber BG, Mateas M, Jhala A (2012) Learning from demonstration for goal-driven autonomy. In: AAAI
64. Wood DJ (1998) How children think and learn: the social contexts of cognitive development. Blackwell, Oxford
65. Wusteman J (1992) Explanation-based learning: a survey. Artif Intell Rev 6:243–262
66. Ye G, Alterovitz R (2011) Demonstration-guided motion planning. In: Proceedings of the international symposium on robotics research (ISRR)

Amit Kumar Pandey is Research Scientist at Aldebaran Robotics, Paris, France. Previously for 6 years he worked as researcher at LAAS-CNRS (French National Center for Scientific Research), Toulouse, France. From there he also obtained his Ph.D. degree in 2012 in Robotics delivered by INSA, University of Toulouse. He is the second prize winner (tie) of the prestigious Georges Giralt Award for the best Ph.D. Thesis in Robotics in Europe, awarded by EU Robotics. With full fellowship, he obtained his MS by Research degree, specialization in Robotics, in 2007 from IIIT-Hyderabad, India. His current research interest includes Human Robot Interaction (HRI), Socially Intelligent Robots, Planning and Learning, and Robots Cognitive Architecture. On these aspects he has been actively contributing in various national and European Union (EU) projects as well as involved in their designing and proposal.

Rachid Alami is Senior Scientist at LAAS-CNRS. He received an engineering diploma in computer science in 1978 from ENSEEIHT, a Ph.D. in Robotics in 1983 from University Paul Sabatier (Toulouse, France) and a Habilitation diriger des recherches in 1996. He contributed and took important responsibilities in several national, European and international research and/or collaborative projects (EUREKA, ESPRIT, IST). His main research contributions fall in the fields of robot control architectures, task and motion planning, manipulation, multi-robot cooperation, humanrobot interaction, and more generally robot decisional autonomy. He has also a substantial experience in robotics system integration and transfer operations.