# A Theory of Explanations for Human-Robot Collaboration

**Mohan Sridharan**
School of Computer Science
University of Birmingham
Birmingham B15 2TT, UK

**Ben Meadows**
Department of Electrical and Computer Engineering
University of Auckland
Auckland 1142, NZ

**Abstract**

To collaborate effectively with humans in complex, dynamic domains, robots need the ability to explain their knowledge, decisions, and experience in human-understandable terms. Towards this objective, this paper makes two contributions. First, we present a theory of explanations that includes claims about representing, reasoning with, and learning knowledge to support the construction of explanations, three fundamental axes used to characterize explanations, and a methodology for constructing these explanations. Second, we describe a cognitive architecture for robots that implements this theory and supports scalability to complex domains and explanations. We demonstrate the working of this architecture in the context of a simulated robot assisting humans by finding and moving desired objects to target locations or people, or by following recipes to bake biscuits.

## 1 Motivation

To collaborate effectively with humans in complex domains such as warehouses and hospitals, it is important for a robot to communicate its beliefs, decisions and experiences in a suitable manner. Despite considerable research, it is challenging to enable a robot to provide such explanations. The robot often makes decisions based on different descriptions of uncertainty and incomplete domain knowledge. For instance, a robot in a university building may know that "books are usually in the library", and infer based on processing sensor inputs that "the robotics book is in Prof. X's office with $90\%$ certainty". While reasoning with this knowledge to compute a plan that achieves a given goal, the robot evaluates different options using different performance measures, e.g., "corridor-1 is a shorter path to the library than corridor-2, but it is likely to be more crowded". In addition, the robot may acquire new knowledge by interacting with humans or the domain, and this information may complement or contradict the existing beliefs. Furthermore, when a human does solicit an explanation, the robot needs to provide the information in a suitable format and at an appropriate level of abstraction for it to be useful.

We seek to formalize the process of explaining enacted or computed plans to achieve a desired goal, the associated knowledge and beliefs, and the experiences that informed these beliefs, in the context of a robot assisting humans. With the increasing use of machine learning and planning algorithms, "explainability" or "interpretability" is also being considered important to establish "trust" in these algorithms. However, generating explanations by just reasoning about assumed or actual action executions, or about the use of axioms governing domain dynamics, may be perceived as unsatisfactory, lacking information, computationally expensive, or likely to contain "too many implementation details" (Johnson 1994b). This paper is a step towards addressing these problems in the context of human-robot collaboration and makes the following contributions:

1. Presents claims about representing, reasoning with, and learning knowledge to support explanations.
2. Characterizes explanations along three axes based on abstraction of representation, explanation specificity, and explanation verbosity, and presents a methodology for constructing explanations.
3. Describes a cognitive architecture for robots that implements the theory of explanations comprising the proposed claims, axes and methodology.

We illustrate the architecture's capabilities in the context of a simulated robot assisting humans by finding and delivering desired objects to target locations or people in an office building, or by following recipes to bake biscuits in a kitchen. We use these examples to show that the theory and architecture scale to complex domains and explanations. We first review related work in Section 2, followed by the theory of explanation and architecture in Section 3. Section 4 describes execution traces that demonstrate the architecture's capabilities, followed by the conclusions and directions for further research in Section 5.

## 2 Related Work

Research in cognition, psychology and linguistics influenced some of the early work on explanations, e.g., Friedman (1974) presented a theory of scientific explanation in terms of generality, objectivity, and connectivity. Grice (1975) characterized a cooperative response as being valid, sufficiently informative, relevant, and unambiguous. The intersection of computing with these fields led to the work on "explanation" tasks, e.g., explaining decisions made by an agent (McKeown and Swartout 1987). Explainable AI is attracting a lot of attention in recent times

due to the increasing use of AI and machine learning (especially deep learning) algorithms in different applications. Although humans do not need every such algorithm to provide detailed explanations, explainability does help establish accountability and trust, and makes it easier to debug the algorithms (Sheh 2017b).

Researchers have used human studies to identify principles governing explanations (Brown and Kleeck 1989), present a theory requiring explanations to be easy to understand, context-specific and justifiable (Gregor and Benbasat 1999), and to emphasize the importance of the right way to present information (Feiner and McKeown 1989). Prior work on agents describing and justifying decisions, e.g., in a tactical air-to-air combat domain (Johnson 1994a), indicates that an agent should describe its activities, goals, rationale and experiences; answer questions; and provide explanations in suitable formats based on a model of user beliefs. Recent work on a framework for explaining the predictions of any classifier (Koh and Liang 2017; Ribeiro, Singh, and Guestrin 2016) also indicates that explanations must be interpretable, responsive to user needs, and model-agnostic.

There is very little work on the kind of recounting (of experiences, plans etc) that we are focusing on, but explanations have been categorized into those of outcomes at the system level ("reasoning trace explanations"), of strategies at the problem-solving level ("strategic explanations"), and of the reasons for states and actions ("deep explanations") (Southwick 1991). Sheh (2017a) distinguishes between three explanation "depths", where model attributes, the use of these attributes, or information about model generation, are considered for explanation; he also categorizes explanations into: teaching, introspective tracing, introspective informative, post-hoc, and execution.

Very few approaches systematically identify the dimensions characterizing explanations. In one recent work, a robot use three axes (abstraction, specificity, locality) to *verbalize* its experience to humans (Rosenthal, Selvaraj, and Veloso 2016). This effort, although interesting, uses hard-coded methods for the specific task of traversing a building. It does not generalize along these axes or to other domains, e.g., locality determines the subset of the route to be provided to the explanation generator instead of using this information for explanation generation; specificity considers the robot's complete route at the first level, the sub-route per floor of a specific building at second level, and so on. The authors claim to derive the three axes from research on user preferences (Dey 2009; Bohus, Saw, and Horvitz 2014; Thomason et al. 2015), but these studies are too dissimilar to the task of an agent narrating its experiences, and thus do not support a general theory of explanations for human-robot collaboration. In prior work, we outlined the ability of agents to explain their decisions and the reasoning that produced these decisions (Langley et al. 2017). We also identified functional capabilities and the key elements of systems designed for explainable agents. Here, we expand on these ideas to provide a theory of explanation in the context of human-robot collaboration.

# 3 Theory of Explanation and Architecture

In this section, we describe our theory of explanation (Section 3.1), followed by an architecture that implements this theory (Section 3.2).

## 3.1 Theory of Explanation

Based on insights gained from prior work, we have identified the following guiding principles or claims for explanations in human-robot collaboration:

1. Explanations should present context-specific information relevant to the task, domain and/or the question under consideration, at an appropriate level of abstraction.
2. Explanations should be able to describe knowledge, beliefs, actions, goals, decisions, rationale for decisions, and underlying strategies or models in real-time.
3. Explanation generation systems should have minimal task-specific or domain-specific components.
4. Explanation generation systems should model and use human understanding and feedback to inform their choices while constructing explanations.
5. Explanation generation systems should use knowledge elements that support non-monotonic revision based on immediate or delayed observations obtained from active exploration or reactive action execution.

Based on these guiding principles, we propose three fundamental axes to characterize explanations:

1. **(Representation abstraction)** This axis models the levels of abstraction at which knowledge is represented for reasoning and explanation. For instance, the robot may use a coarse-resolution domain description in terms of rooms and the objects (e.g., cups, books) in these rooms, or it may use a fine-resolution description in terms of grid cells in the rooms and object parts (e.g., cup handle, cup base) in these grid cells.
2. **(Communication specificity)** This axis models what the robot focuses on while communicating with the human. For instance, to explain the decision to traverse a longer corridor instead of a shorter corridor, the robot may provide a: (i) brief explanation that considers the crowdedness of the corridors; or (ii) an elaborate explanation that considers the crowdedness of the corridors, the robot's energy levels and ability to move safely, and the objective of maximizing task completion and safety.
3. **(Communication verbosity)** This axis models the comprehensiveness of the response provided. For instance, when asked to explain the plan computed to achieve a particular goal, the robot may describe: (i) just the last action in its plan and how it achieves the goal; (ii) all the the actions in the plan that results in the goal being achieved; or (iii) all the actions along with the preconditions and effects of each of them to show how the goal is achieved.

We also propose the following methodology to provide explanations in response to questions from human users:

1. Choose a suitable position along each of three axes to provide explanations in response to a specific question.

2. Determine what needs to be described in the explanation. This may take the form of one or more of knowledge elements, beliefs, actions, goals, choices, and justification for these choices.
3. Produce relevant, context-specific explanations that limit the use of domain-specific knowledge. Construct verbalizations of these explanations to answer the user query.
4. Use human feedback to revise the choice in Step-1.

We next describe an architecture that implements this theory and discuss its implications.

### 3.2 Cognitive Architecture

Figure 1 shows our overall architecture that reasons with tightly-coupled transition diagrams at different resolutions. Depending on the domain and tasks at hand, the robot chooses to plan and execute actions at two specific resolutions, but constructs explanations at other resolutions as needed. For ease of explanation, we will focus on two resolutions, with the fine-resolution transition diagram defined as a *refinement* of the coarse-resolution transition diagram; we will discuss the extension to additional resolutions later in the paper. For any given goal, non-monotonic logical reasoning with commonsense domain knowledge in the coarse resolution provides a plan of *abstract actions*. Each abstract transition is implemented as a sequence of concrete actions by *zooming* to and reasoning with the relevant part of the fine-resolution transition diagram. Each concrete action is executed using probabilistic models of the uncertainty in perception and actuation, with the outcomes added to coarse-resolution history. Reasoning with commonsense knowledge also informs and guides the interactive learning of previously unknown actions, action capabilities and axioms. The architecture thus combines the complementary strengths of declarative programming, probabilistic reasoning, and relational learning, and may be viewed as a logician and statistician working together. Some of these components have been described in other papers (Gomez, Sridharan, and Riley 2018; Sridharan et al. 2018; Sridharan and Meadows 2018). We focus on explanations and briefly describe all components using the following example domain.

**Example Domain 1.** *[Robot Assistant (RA)]*
A robot finds and delivers objects to people or places ($study$, $office$, $workshop$, $kitchen$) in an indoor domain. Each place may have instances of objects such as $book$ and $cup$. Each human has a $role$ (e.g., $engineer$, $manager$, $sales$). Objects have a $size$ and $color$. Some other details include:

- The position of the robot and objects can change.
- The robot can move to a place, pick up or put down an object, or deliver an object to a person.
- The domain may be viewed at different resolutions, e.g., a place can be one or four rooms or one of four cells within each room, and the robot may move an object to particular rooms or particular cells.

Reasoning occurs in finite time steps with partial knowledge of rules governing the domain dynamics, e.g., objects can only be delivered to people in the same place as the robot.

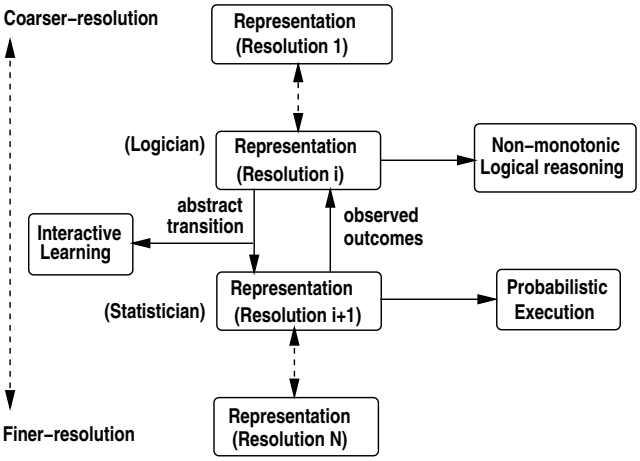


Figure 1: Architecture supports representation and reasoning with tightly coupled transition diagrams at different resolutions. It combines the complementary strengths of declarative programming and probabilistic reasoning.

**Action Language** Action languages are formal models of parts of natural language used for describing transition diagrams of dynamic systems. Our architecture uses action language $\mathcal{AL}_d$ (Gelfond and Inclezan 2013) to describe the transition diagrams at different resolutions. $\mathcal{AL}_d$ has a sorted signature with *statics*, i.e., domain attributes whose truth values cannot be changed by actions, *fluents*, i.e., domain attributes whose truth values can be changed by actions, and *actions*, a set of elementary operations. Fluents can be *basic*, which obey inertia laws and can be changed by actions, or *defined*, which do not obey the laws of inertia and are not changed directly by actions. A domain attribute or its negation is a *literal*. $\mathcal{AL}_d$ allows three types of statements: *causal law*, *state constraint* and *executability condition*.

**Knowledge Representation** The coarse-resolution domain description comprises a system description $\mathcal{D}_c$ of transition diagram $\tau_c$, which is a collection of statements of $\mathcal{AL}_d$, and history $\mathcal{H}_c$. $\mathcal{D}_c$ includes a sorted signature $\Sigma_c$ and axioms. For the RA domain, $\Sigma_c$ defines basic sorts such as $place$, $thing$, $entity$, $robot$, $person$, $object$, $cup$ and $book$, arranged hierarchically, e.g., $object$ and $robot$ are subsorts of $thing$, a sort $step$ for temporal reasoning, and instances of sorts, e.g., $rob_1$, $cup_1$, $book_2$. For the RA domain, $\Sigma_c$ includes statics such as $next\_to(place, place)$ and $obj\_color(object, color)$, fluents such as $loc(thing, place)$ and $in\_hand(robot, object)$ and actions $move(robot, place)$, $pickup(robot, object)$, $putdown(robot, object)$, and $give(robot, object, person)$; we can also include exogenous actions to explain unexpected observations. $\Sigma_c$ also includes $holds(fluent, step)$ to imply that a particular fluent is true at a particular time step. Next, $\mathcal{D}_c$ for the RA domain includes axioms such as:

$move(rob_1, P)$ **causes** $loc(rob_1, P)$
$loc(O, P)$ **if** $loc(rob_1, P)$, $in\_hand(rob_1, O)$
**impossible** $give(rob_1, O, P)$ **if** $loc(rob_1, L_1) \neq loc(P, L_2)$

that describe the domain dynamics and are used for planning and diagnostics. Finally, the history $\mathcal{H}_c$ expands the typical definition, which is a record of fluents observed to be true or false at a particular time step, $obs(fluent, boolean, step)$, and the occurrence of an action at a particular time step, $occurs(action, step)$, to represent defaults describing the values of fluents in the initial state. For instance, $\mathcal{H}_c$ of the RA domain encodes the statement "books are usually in the library and if it not there, they are normally in the office" and the exception "cookbooks are in the kitchen". For more details, including the model of history with initial state defaults, please see (Sridharan et al. 2018).

**Reasoning with Knowledge** Reasoning is performed by translating the domain description to a program in CR-Prolog, a variant of Answer Set Prolog (ASP) that incorporates consistency restoring (CR) rules (Balduccini and Gelfond 2003); we use the terms CR-Prolog and ASP interchangeably in this paper. ASP is based on stable model semantics, and supports *default negation* and *epistemic disjunction*, e.g., unlike "$\neg a$" that states *a is believed to be false*, "$not\ a$" only implies *a is not believed to be true*, i.e., each literal can be true, false or "unknown". ASP represents recursive definitions and constructs that are difficult to express in classical logic formalisms, and supports nonmonotonic logical reasoning. For coarse-resolution reasoning, the program $\Pi(\mathcal{D}_c, \mathcal{H}_c)$ includes the signature and axioms of $\mathcal{D}_c$, inertia axioms, reality checks, closed world assumptions for defined fluents and actions, and observations, actions, and defaults from $\mathcal{H}_c$. Every default also has a CR rule that allows the robot to assume the default's conclusion is false to restore consistency under exceptional circumstances. An *answer set* of $\Pi$ represents the set of beliefs of the robot associated with $\Pi$. Algorithms for computing entailment, and for planning and diagnostics, reduce these tasks to computing answer sets of CR-Prolog programs. We compute answer sets using the SPARC system (Balai, Gelfond, and Zhang 2013).

**Refinement, Zooming and Probabilistic Execution** For any given goal, the plan of abstract actions obtained by reasoning with $\Pi(\mathcal{D}_c, \mathcal{H}_c)$ cannot be executed directly. To implement these abstract actions, we construct a fine-resolution system description $\mathcal{D}_f$ of transition diagram $\tau_f$ that is a *refinement* of, and is tightly coupled to, $\mathcal{D}_c$. Refinement may be viewed as looking through a magnifying lens, potentially discovering domain structures that were previously abstracted away. We only briefly describe refinement due to space limitations; see (Sridharan et al. 2018) for details.

We first construct the *weak refinement* ignoring the robot's ability to observe the values of fluents. Signature $\Sigma_f$ includes (i) elements of $\Sigma_c$; (ii) new sort for every sort of $\Sigma_c$ magnified by the increase in resolution; (iii) counterparts for each magnified domain attribute of $\Sigma_c$ and actions with magnified sorts; and (iv) domain-dependent statics $component(O^*, O)$ relating magnified objects and their counterparts. For the RA domain, basic sorts in $\Sigma_f$ include:

$$place^* = \{c_1, \ldots, c_m\},\ cup^* = \{cup_1\_base, cup_1\_handle\}$$

where $\{c_1, \ldots, c_m\}$ are cells in places, $base$ and $handle$ are components of $cup$, and "*" represents fine-resolution counterparts. New domain attributes and actions of $\Sigma_f$ include:

$$loc^*(thing^*, place^*),\ next\_to^*(place^*, place^*)$$
$$move^*(robot, place^*),\ \ in\_hand^*(robot, cup^*)$$

Axioms of $\mathcal{D}_f$ are obtained by restricting the axioms of $\mathcal{D}_c$ to $\Sigma_f$, e.g., axioms of the RA domain include:

$$move^*(R, C)\ \textbf{causes}\ loc^*(R, C)$$
$$loc(O, P)\ \textbf{if}\ component(C, P),\ loc^*(O, C)$$

Next, our *theory of observation* expands $\Sigma_f$ to include *knowledge-producing* action $test(robot, fluent)$ that checks the value of fluents and changes the value of *knowledge fluents* that describe observations of fluents. Axioms are added to $\mathcal{D}_f$ to encode the $test$ actions, using suitable domain-dependent defined fluents, e.g., to describe when the robot can test the value of particular fluents. For each transition between coarse-resolution states $\sigma_1$ and $\sigma_2$, there is a path in $\tau_f$ between a refinement of $\sigma_1$ and a refinement of $\sigma_2$—the proof is in (Sridharan et al. 2018).

Although $\mathcal{D}_f$ does not have to be revised unless the domain changes significantly, reasoning with $\mathcal{D}_f$ becomes computationally unfeasible for complex domains. In our architecture, for each abstract transition $T = \langle \sigma_1, a^H, \sigma_2 \rangle \in \tau_H$, the robot automatically *zooms* to and reasons with $\mathcal{D}_f(T)$, the part of $\mathcal{D}_f$ *relevant* to $T$. To obtain $\mathcal{D}_f(T)$, the robot determines the object constants of $\Sigma_c$ relevant to $T$, restricts $\mathcal{D}_c$ to these object constants to obtain $\mathcal{D}_c(T)$, computes the basic sorts of $\Sigma_f(T)$ as those of $\Sigma_f$ that are components of the basic sorts of $\mathcal{D}_c(T)$, restricts domain attributes and actions of $\Sigma_f(T)$ to these basic sorts, and restricts axioms of $\mathcal{D}_f$ to $\Sigma_f(T)$. In the RA domain, if $T = \langle \sigma_1, move(rob_1, kitchen), \sigma_2 \rangle$ with $loc(rob_1, office) \in \sigma_1$, the basic sorts of $\Sigma_f(T)$ include $robot = \{rob_1\}$, $place = \{office, kitchen\}$ and $place^* = \{c_i : c_i \in kitchen \cup office\}$. Domain attributes include $loc^*(rob_1, C)$ taking values from $place^*$, $loc(rob_1, P)$ taking values from $place$ etc, and actions include $move^*(rob_1, c_i)$ and suitable $test$ actions. Restricting the axioms of $\mathcal{D}_f$ to $\Sigma_f(T)$ removes axioms for $pickup$ and $putdown$, and irrelevant constraints. For any coarse-resolution transition $T$, there is a path in $\mathcal{D}_f(T)$ between a refinement of $\sigma_1(T)$ and a refinement of $\sigma_2(T)$—see (Sridharan et al. 2018) for the proofs.

Our prior work constructed a partially observable Markov decision process from $\mathcal{D}_f(T)$ to implement $T$. Since this is computationally inefficient, we construct and solve $\Pi(\mathcal{D}_f(T), \mathcal{H}_f)$ to obtain a sequence of concrete actions, each of which is executed using existing algorithms (e.g., for path planning and object recognition) based on probabilistic models of the uncertainty in sensing and actuation. High-probability outcomes of a concrete action are elevated to statements with complete certainty in $\mathcal{H}_f$, and the outcomes of reasoning with $\Pi(\mathcal{D}_f(T), \mathcal{H}_f)$ are added to $\mathcal{H}_c$.

**Interactive Learning** Reasoning with incomplete domain knowledge can result in incorrect or suboptimal outcomes. The robot can learn previously unknown actions and related axioms, but doing so in the most generic form may require many labeled examples, which is not always feasible in

robot domains. Also, humans may not have the time and expertise to provide labeled examples, and an action's effects may be immediate or delayed. For interactive acquisition of labeled examples and knowledge, our architecture has two schemes: (i) active learning of actions and causal laws from human verbal descriptions of actions of other robots; and (ii) cumulative learning of action capabilities (i.e., affordances) and axioms using decision tree induction and relational reinforcement learning based on observations from active exploration or reactive action execution. For complete details, please see (Sridharan and Meadows 2018).

**Constructing Explanations** To construct an explanation in response to a query, the robot builds on the methodology in Section 3.1. Existing implementations of algorithms enable the robot to identify objects, actions and relations, understand parts of speech and a controlled vocabulary, construct sentences from templates based on the controlled vocabulary, distinguish between physical entities and mental concepts, and to solicit feedback from humans. The steps to be followed are:

1. Parse input query to extract cues (e.g., part of speech and words in vocabulary) indicating the objects, actions, and relations of interest.
2. Select suitable point along the representation abstraction axis. This is the resolution used for commonsense reasoning or action execution unless user query indicates otherwise. Reuse point along this axis used in the most recent interaction unless query indicates otherwise.
3. Choose points along specificity and verbosity axes based on cues from query. Use these selections to translate object references to descriptions. This includes the choice of object attributes to use as modifiers, e.g., "a room" or "a medium-sized, library room", and the choice of reference to relevant knowledge, e.g., "a library room", "the library room", or $study_1$ all refer to the same place.
4. Reason with domain knowledge to identify *relevant* knowledge elements (objects, actions, relations etc). Transform elements to text descriptions using controlled vocabulary and domain knowledge templates, e.g., $pick\_up(rob_1, book_2)$, where $book_2$ is robotics book, provides "the robot picked up the robotics book".

The choices made and the domain's quantization influences the language and ambiguity of the explanations, e.g., high verbosity and high specificity descriptions are unambiguous whereas low verbosity and low specificity descriptions are confusing, and if rooms have $10 \times 10$ cells instead of $2 \times 2$ cells, the number of actions to achieve a goal and the length of the explanations increase. The software implementation of the control loop and the construction of explanations is available in our repository (Meadows and Sridharan 2018).

## 4 Execution Examples

Our architecture's ability to provide explanations is illustrated using Example Domain 1, a variant of this domain ($RA^*$) that explores the impact of quantization on explanation, e.g., each room may have up to 100 cells (i.e., $10 \times 10$ grid) instead of four cells (i.e., $2 \times 2$ grid), and the following domain based on the scenario in (Bollini et al. 2013).

**Example Domain 2.** *[Robot Baker ($RB$)]*
A robot baker in a kitchen has two work tables, one for preparation and another with a toaster oven. For an item to be baked, all ingredients ($cocoa$, $sugar$, $flour$, $cornflakes$, and $butter$) are pre-measured and placed in bowls on the table. Kitchen $tools$ are characterized by $type$ ($bowl$, $tray$, $oven$), $material$ ($plastic$, $metal$), $size$ ($small$, $medium$, $large$) and $color$ ($red$, $yellow$, $silver$), e.g, five plastic ingredient bowls of various sizes and colors, a large mixing bowl, a metal oven tray, and a toaster oven. Other details include:

- The robot has grasping and stirring manipulators.
- The domain may be viewed at different resolutions, e.g., the tools may be on the work table or in one of its six cells.

This domain's encoding involves deeper sort hierarchies than the $RA$ domain, e.g., an object may be a $mixing\ bowl$, which is a $bowl$, which is a $container$, which is an $object$, which is a $thing$. Also, plans in the domain, which represent recipes being followed, can be more varied, with many more coarse and fine-resolution actions, e.g., to bake "Afghan biscuits", the robot has to pour, mix, scrape, preheat, reposition, bake, etc, each of which can be represented by up to ten fine-resolution actions.

Now, consider the following execution traces of explanations. For ease of understanding, we omit some parts of the explanations, depicting them as "[...]" in the text. We begin with an example in the $RA$ domain.

**Execution Example 1.** *[Explanations in $RA$ Domain]*
Consider a robot in the RA domain that has just moved a book to the study and delivered a book from the study to the engineer in the office.

- **Human:** "Please describe the plan you executed."
  **Robot:** " [...] I moved to the $study$. I put down a small programming book on $table_1$. I picked up a large robotics book from $table_1$. I moved to the $office$. I delivered the large robotics book to the $engineer$ in the $office$. I achieved the goal."
- When asked to provide a more detailed description, the robot revises the operating point along the abstraction axis; in the current implementation, the fine-resolution description is used to generate an explanation.
  **Human:** "Provide a more detailed description."
  **Robot:** "[...] I moved to cell $c_2$ of the $study$, the room at the end of the corridor. [...] I moved to cell $c_4$ in the $study$ and put down a small programming book with a hard cover on the table. I picked up the large robotics book with a soft cover by its spine from the table. [...] I moved to cell $c_2$ of the $study$. [...] I moved to cell $c_7$ of the $office$. I gave the large robotics book to the $engineer$. I achieved the goal."
- During planning or plan execution, the robot explains the need for particular actions, and the direct and indirect effects of actions.
  **Human:** "Why did you move to cell $c_4$ of the $study$?"

**Robot:** "I had to put the small programming book on the table. The table is in cell $c_4$ of the $study$."

- **Human:** "Relationally describe the effects of moving to cell $c_4$ of the $study$?"
  **Robot:** $loc^*(rob_1, cell_4), loc(rob_1, study)$.

Now, consider the effect of increasing the quantization, e.g., the $RA^*$ domain with a $10 \times 10$ grid of cells in each room instead of the $2 \times 2$ grid in the $RA$ domain. This increase in quantization causes a significant increase in the number of steps in the plans and in any explanation of the plans, especially in the fine resolution. For example, there are $\approx 1000$ steps in the explanation of the same plan described above. However, as a result of refinement and zooming, there is only a linear increase in the planning time and the robot is able to provide explanations in real time. Similar results were obtained with other execution scenarios in the $RA$ domain. Next consider an execution trace in the $RB$ domain.

**Execution Example 2.** *[Explanations in RB Domain]* Consider the interaction of a robot baker with a human in the $RB$ domain. The robot has just completed a baking task.

- **Human:** "Please describe the plan you executed."
  **Robot:** " [...] I poured the $cocoa$ in the yellow small ingredient bowl to a red large mixing bowl. I poured the $cornflakes$ in a red medium ingredient bowl to the red large mixing bowl. I mixed the contents of the red large mixing bowl. I scraped the mixture in the red large mixing bowl to a metallic medium baking tray. I picked up the silver metallic medium baking tray. I moved to $table_2$ with the silver metallic medium baking tray in my hand. I opened the silver toaster oven on $table_2$. I put the silver metallic medium baking tray in the silver toaster oven on $table_2$. I closed the silver toaster oven. I baked the silver metallic medium baking tray in the silver toaster oven. I waited for 20 minutes. I opened the silver toaster oven on $table_2$. I picked up the silver metallic medium baking tray from the silver toaster oven. I put the silver metallic medium baking tray on $table_2$. [...] I achieved my goal."
- When asked to provide more concise and relevant information, the robot moves its operating point along the specificity and verbosity axes.
  **Human:** "Provide a concise and relevant description."
  **Robot:** "[...] I poured the $cocoa$ in $bowl_5$ to $mixing\_bowl_1$. I poured the $cornflakes$ in $bowl_2$ to $mixing\_bowl_1$. I mixed the contents of $mixing\_bowl_1$. I scraped the mixture in $mixing\_bowl_1$ to $baking\_tray_1$. I picked up $baking\_tray_1$. I moved to $table_2$. I opened $toaster\_oven_1$ on $table_2$. I put $baking\_tray_1$ in $toaster\_oven_1$. I baked $baking\_tray_1$ in $toaster\_oven_1$ for 20minutes. I opened $toaster\_oven_1$. I picked up $baking\_tray_1$. I put $baking\_tray_1$ on $table_2$. [...] I achieved my goal."
- **Human:** "Please be more concise."
  **Robot:** "I preheated $toaster\_oven_1$. I mixed $flour$, $butter$, $sugar$, $cocoa$, and $cornflakes$ in $mixing\_bowl_1$. I scraped the mixture to $baking\_tray_1$. I baked $baking\_tray_1$ in $toaster\_oven_1$ for 20 minutes. I put $baking\_tray_1$ on $table_2$. I achieved my goal."
- The robot explains the need for, and can potentially explain the likelihood of, particular actions and outcomes during planning or execution.
  **Human:** "Why did you move the baking tray to $table_2$?"
  **Robot:** "I need to put the baking tray in the toaster oven that is on $table_2$."
  **Human:** "How likely is it that there is $cocoa$ in the small yellow ingredient bowl?"
  **Robot:** "I am $95\%$ sure there is no $cocoa$ left in the small yellow ingredient bowl."

Similar results were obtained for other scenarios and questions in the domains considered in this paper.

## 5 Discussion and Future Work

The ability to explain its beliefs, decisions and experience is important for a robot collaborating with humans. In this paper, we first presented a theory of explanations comprising (i) claims about representing, reasoning with, and learning knowledge to support effective explanations; (ii) three axes to characterize explanations; and (iii) a methodology for constructing explanations. This theory is motivated by insights from existing studies and our work on designing cognitive architectures for human-robot collaboration. Next, we described a cognitive architecture for knowledge representation, reasoning and learning, which also implements the proposed theory. We described execution traces illustrating the automatic construction of suitable explanations in response to queries from humans. Although we focused on explanations in this paper, the overall architecture uses tightly-coupled transition diagrams at different resolutions to support non-monotonic logical reasoning with commonsense knowledge and probabilistic reasoning with quantitative models of the uncertainty in sensing and actuation.

The proposed theory and architecture open up many directions for further research. First, this paper focused on one coarse-resolution and one fine-resolution description for ease of explanation. However, other experiments (not reported here) indicate that the definitions of refinement, zooming and relevance used here readily apply to additional resolutions as well. The tight coupling between the resolutions result in smooth transfer of information and control between the different resolutions. In future work, we will more thoroughly explore the automatic transition between multiple resolutions on demand, constructing explanations at the level of abstraction desired by the user the robot is interacting with. Second, our current architecture does not yet provide partial explanations or revise the operating point along the three axes with respect to only part of the observations (or history) being explained. It is possible to provide such partial explanations by limiting reasoning to the desired part of the history. Third, we will conduct studies with human subjects to evaluate the effectiveness and usability of our theory and architecture. These studies will also provide important feedback that can be used to revise the claims, methodology and the representation encoded in the architecture. Finally, we will also explore the extension of this architecture to teams of robots and humans collaborating to achieve a shared objective in complex domains.

## References


Balai, E.; Gelfond, M.; and Zhang, Y. 2013. Towards Answer Set Programming with Sorts. In *International Conference on Logic Programming and Nonmonotonic Reasoning*.

Balduccini, M., and Gelfond, M. 2003. Logic Programs with Consistency-Restoring Rules. In *AAAI Spring Symposium on Logical Formalization of Commonsense Reasoning*, 9–18.

Bohus, D.; Saw, C.; and Horvitz, E. 2014. Directions Robot: In-the-wild Experiences and Lessons Learned. In *International Conference on Autonomous Agents and Multiagent Systems*, 637–644. International Foundation for Autonomous Agents and Multiagent Systems.

Bollini, M.; Tellex, S.; Thompson, T.; Roy, N.; and Rus, D. 2013. Interpreting and Executing Recipes with a Cooking Robot. In J. Desai, G. Dudek, O. K., and Kumar, V., eds., *Experimental Robotics, Springer Tracts in Advanced Robotics*, volume 88. Springer, Heidelberg. 481–495.

Brown, R., and Kleeck, M. H. V. 1989. Enough Said: Three Principles of Explanation. *Journal of Personality and Social Psychology* 57(4):590–604.

Dey, A. K. 2009. Explanations in context-aware systems. In *Proceedings of the Fourth International Conference on Explanation-Aware Computing*, 84–93. AAAI Press.

Feiner, S. K., and McKeown, K. R. 1989. Coordinating Text and Graphics in Explanation Generation. In *Proceedings of the 1989 Workshop on Speech and Natural Language*, 424–433. Association for Computational Linguistics.

Friedman, M. 1974. Explanation and scientific understanding. *The Journal of Philosophy* 71(1):5–19.

Gelfond, M., and Inclezan, D. 2013. Some Properties of System Descriptions of $AL_d$. *Journal of Applied Non-Classical Logics, Special Issue on Equilibrium Logic and Answer Set Programming* 23(1-2):105–120.

Gomez, R.; Sridharan, M.; and Riley, H. 2018. Representing and Reasoning with Intentional Actions on a Robot. In *ICAPS Workshop on Planning and Robotics (PlanRob)*.

Gregor, S., and Benbasat, I. 1999. Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS Quarterly* 497–530.

Grice, H. P. 1975. Logic and Conversation. In Cole, P., and Morgan, J., eds., *Syntax and semantics*. New York: Academic Press. 41–58.

Johnson, W. L. 1994a. Agents that explain their own actions. In *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*, 87–95.

Johnson, W. L. 1994b. Agents that learn to explain themselves. In *Proceedings of the Twelfth AAAI National Conference on Artificial Intelligence*, 1257–1263. AAAI Press.

Koh, P. W., and Liang, P. 2017. Understanding Black-box Predictions via Influence Functions. In *International Conference on Machine Learning (ICML)*, 1885–1894.

Langley, P.; Meadows, B.; Sridharan, M.; and Choi, D. 2017. Explainable Agency for Intelligent Autonomous Systems. In *Innovative Applications of Artificial Intelligence (IAAI)*.

McKeown, K. R., and Swartout, W. R. 1987. Language generation and explanation. *Annual Review of Computer Science* 2(1):401–449.

Meadows, B., and Sridharan, M. 2018. Software implementing theory of explanation. `https://github.com/bmeadows/Theory-of-explanations/`.

Ribeiro, M.; Singh, S.; and Guestrin, C. 2016. "Why should I trust you?" Explaining the Predictions of any Classifier. In *Proceedings of the Twenty-Second ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. ACM.

Rosenthal, S.; Selvaraj, S.; and Veloso, M. 2016. Verbalization: Narration of Autonomous Robot Experience. In *Twenty-Fifth International Joint Conference on Artificial Intelligence*, 862–868. AAAI Press.

Sheh, R. 2017a. Different XAI for different HRI. In *AAAI Fall Symposium on Artificial Intelligence for Human-Robot Interaction (Technical Reports)*, 114–117.

Sheh, R. 2017b. "Why did you do that?" Explainable Intelligent Robots. In *AAAI Workshop on Human-Aware Artificial Intelligence*, 628–634.

Southwick, R. W. 1991. Explaining reasoning: An overview of explanation in knowledge-based systems. *The Knowledge Engineering Review* 6(1):1–19.

Sridharan, M., and Meadows, B. 2018. Knowledge Representation and Interactive Learning of Domain Knowledge for Human-Robot Collaboration. *Advances in Cognitive Systems* 7:77–96.

Sridharan, M.; Gelfond, M.; Zhang, S.; and Wyatt, J. 2018. REBA: A Refinement-Based Architecture for Knowledge Representation and Reasoning in Robotics. Technical report, `http://arxiv.org/abs/1508.03891`.

Thomason, J.; Zhang, S.; Mooney, R. J.; and Stone, P. 2015. Learning to interpret natural language commands through human-robot dialog. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 1923–1929. AAAI Press.