

# Towards a Theory of Explanations for Human-Robot Collaboration

Mohan Sridharan · Ben Meadows

Received: date / Accepted: date

**Abstract** This paper makes two contributions towards enabling a robot to provide explanatory descriptions of its decisions, the underlying knowledge and beliefs, and the experiences that informed these beliefs. First, we present a theory of explanations comprising (i) claims about representing, reasoning with, and learning domain knowledge to support the construction of explanations; (ii) three fundamental axes to characterize explanations; and (iii) a methodology for constructing these explanations. Second, we describe an architecture for robots that implements this theory and supports scalability to complex domains and explanations. We demonstrate the architecture’s capabilities in the context of a simulated robot (a) moving target objects to desired locations or people; or (b) following recipes to bake biscuits.

**Keywords** Human-Robot Collaboration · Explanations · Non-monotonic Logical Reasoning · Probabilistic Planning

## 1 Motivation

Robots can collaborate more effectively with humans if they can describe their decisions, the underlying beliefs, and the experiences that informed these beliefs. Enabling a robot to provide such explanatory descriptions is a challenging problem. 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 may process sensor inputs to infer that “the robotics book is in Prof. X’s office with 90% certainty”. While reasoning with this knowl-

edge to compute a plan for any given goal, e.g., “move the robotics book to the meeting room”, the robot evaluates the available options using different performance measures, e.g., “corridor-1 is a shorter path to the library than corridor-2, but it is more crowded”. While reasoning or executing actions in the domain, the robot acquires new information that may complement or contradict existing beliefs. Also, human participants (if any) may not have the time or expertise to provide comprehensive information or extensive supervision. Furthermore, when an explanation is solicited by a human, it must be provided in an appropriate format and level of abstraction for it to be useful.

With the increasing use of AI and machine learning algorithms in different applications, there is renewed interest in understanding the decisions of these algorithms as a means to improve the algorithms and promote accountability. There is considerable work on making the decisions of an existing learned model or reasoning system more interpretable, and on modifying an existing learning or reasoning system to make decisions that are easier for humans to understand. Many such approaches tend to be computationally expensive, or are perceived as lacking information or containing too many unnecessary details (Johnson, 1994b). In this paper, we instead seek to formalize a holistic view of the process of describing decisions, beliefs, and experiences during reasoning, learning, and execution in human-robot collaboration. In our formalism, the desired transparency in decision making is fully integrated with, and strongly influenced by, the underlying knowledge representation, reasoning, and learning methods. We make the following contributions:

1. Present a theory of explanations comprising claims about representing, reasoning with, and learning knowledge to support explanations; axes characterizing explanations based on abstraction of representation, explanation specificity, and explanation verbosity; and a methodology for constructing explanatory descriptions.

---

Mohan Sridharan  
School of Computer Science, University of Birmingham, UK  
E-mail: m.sridharan@bham.ac.uk

Ben Meadows  
Department of Electrical and Computer Engineering, University of  
Auckland, NZ  
E-mail: bmea011@aucklanduni.ac.nz

2. Describe an architecture for robots that implements our theory, exploits the underlying representation and reasoning mechanisms to identify and compute the information *relevant* to the human query, and reliably and efficiently constructs suitable explanatory descriptions that answer the query.

We illustrate the architecture’s capabilities in the context of a simulated robot assisting humans by (i) delivering objects to different locations or people in an office building; or (ii) following recipes to bake biscuits in a kitchen. We first review related work (Section 2). We then describe our theory of explanation (Section 3) and our architecture that implements this theory (Section 4). Section 5 explores the impact of our theory, in conjunction with the representation and reasoning methods, on scalable construction of explanatory descriptions, followed by a discussion of future work in Section 6.

## 2 Related Work

Research in cognition, psychology, and linguistics influenced some of the early work on representing and reasoning about explanations. Friedman (1974) presented a theory of scientific explanation in terms of generality, objectivity, and connectivity, and Grice (1975) characterized cooperative response as being valid, informative, relevant, and unambiguous. Fundamental computational models have also been developed for explanation generation (de Kleer et al., 1992; McKeown and Swartout, 1987; Reiter, 1987).

With AI and machine learning algorithms being used in different applications, there is renewed interest in understanding their outcomes as a means to improve reliability and establish accountability. Workshops and sessions have been organized at premier conferences on topics such as *Explainable AI* and *Explainable Planning* in the last few years. Work in this area can be broadly categorized into two groups (Miller, 2017). Methods in the first group modify or map learned models or reasoning systems to make their decisions interpretable, e.g., explaining the predictions of any classifier by using its decisions to learn an interpretable model (Koh and Liang, 2017; Ribeiro et al., 2016), or adding bias in a planning system towards making decisions easier for humans to understand (Zhang et al., 2017). Methods in the second group present descriptive explanations of the decisions made by reasoning systems, e.g., methods that explain changes in an agent’s goals (Dannenhauer et al., 2018) or plans (Roberts et al., 2018), or allow humans to poll the system about alternative plans (Borgo et al., 2018; Sreedharan et al., 2018). Much of this research has been agnostic to how an explanation is structured and presented (Chakraborti et al., 2017; Winikoff et al., 2018; Borgo et al., 2018; Sreedharan et al., 2018), assumed complete domain knowledge (Chakraborti et al., 2017), or has had limited instantiation in working systems (Roberts et al., 2018; Wicaksono et al.,

2017). Our work is more similar to those in the second group and addresses their limitations.

Human studies have been used to identify principles governing explanations (Brown and Kleck, 1989) and present a theory requiring explanations to be understandable, context-specific, and justifiable (Gregor and Benbasat, 1999). Human studies have also been used to emphasize the importance of presenting information in the right way (Feiner and McKeown, 1989). Prior work on agents describing decisions in a simulated tactical combat domain indicates that an agent should describe its activities, goals, rationale, and experiences; and answer explanatory questions in suitable formats based on a model of user beliefs (Johnson, 1994a).

There is limited work on the kind of recounting (of decisions, underlying beliefs, and experiences that informed these beliefs) that is the focus of our work, but explanations have been grouped into those of outcomes at the system level (“reasoning trace explanations”), strategies at the problem-solving level (“strategic explanations”), and of reasons for particular states and actions (“deep explanations”) (Southwick, 1991). Sheh (2017) distinguishes between three explanation “depths”, where model attributes and their use, or information about model generation, are considered for generating explanations categorized as teaching, introspective tracing, introspective informative, post-hoc, and execution.

Very few approaches systematically identify dimensions suitable for characterizing explanations in human-robot collaboration. In one approach, a robot uses three axes (abstraction, specificity, locality) to *verbalize* its navigation experience to humans (Rosenthal et al., 2016). This work uses methods hard-coded for traversing a building; it does not generalize to other domains. For instance, locality determines the subset of the route to be used to construct the explanation, and specificity considers different parts of the route at different levels. The authors derive these axes from research on user preferences (Dey, 2009; Bohus et al., 2014; Thomason et al., 2015), but these studies are too dissimilar to an agent narrating its experiences. A recent survey of work on explainable agents and robots indicates the need for a general theory of explanations for human-robot collaboration that is integrated with the underlying representation, reasoning and learning abilities (Anjomshoae et al., 2019).

Our prior work outlined the capabilities and systems an agent needs to explain its decisions (Langley et al., 2017). In this paper, we provide a holistic formalization of the process of providing explanatory descriptions of decisions, beliefs, and experiences. We present a theory of explanations for human-robot collaboration that is fully integrated with, and strongly influenced by, the knowledge representation, reasoning, and learning capabilities. We also describe an implementation of this theory in an architecture that supports scalable reasoning. An initial version of this work appeared as a symposium paper (Sridharan and Meadows, 2019).

### 3 Theory of Explanation

This section describes our theory of explanation comprising the guiding principles or claims (Section 3.1), the axes characterizing explanations (Section 3.2), and the methodology for generating explanations (Section 3.3).

#### 3.1 Guiding Principles

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

1. Explanations should present context-specific information relevant to the domain, task or question under consideration, at an appropriate level of abstraction.
2. Explanations should be able to provide online descriptions of decisions, rationale for decisions, knowledge, beliefs, experiences that informed the beliefs, and underlying strategies or models.
3. Explanation generation systems should have as few task-specific or domain-specific components as possible.
4. Explanation generation systems should consider 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.

The implementation of these principles in an architecture influences and is influenced by how knowledge is represented, reasoned with, and learned in the architecture. We choose to expand our prior architecture (Section 4) because it provides capabilities that facilitate this implementation.

#### 3.2 Characteristic Axes

Based on these claims, we propose the use of the following 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 one, the robot may provide: (i) an explanation that considers the corridors' crowdedness; or (ii) an 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 in the plan, along with the preconditions and effects of each of them, to show how the goal is achieved.

Each explanation maps to a point along each of these axes, i.e., it maps to the three-dimensional space defined by these axes. Varying the point along these axes changes the information included in (and communicated by) the explanation, and the format in which this information is communicated.

#### 3.3 Methodology for Generating Explanations

Given an implementation of the claims and the characteristic axes, we propose the following methodology to provide explanations in response to any particular query:

1. In response to a specific question/request, parse human input to determine what is being asked.
2. Choose a suitable position along each of the three axes to inform how the explanation will be structured.
3. Determine what needs to be described in the explanation. This may take the form of choices made, justification for these choices, knowledge elements, beliefs, and experiences that informed these beliefs.
4. Reason with domain knowledge to compute required information (if needed) and to identify relevant knowledge elements. Use the decisions about the structure of the explanation to transform these knowledge elements into context-specific explanatory elements.
5. Construct explanations from the explanatory elements, limiting the use of domain-specific knowledge. Construct verbalizations of these explanations to answer user queries.
6. Use human feedback to revise the choice made in Step 2 about a suitable point along the three axes.

Following this methodology will enable the robot to provide explanations that are relevant to the task and user under consideration. In Section 4.6, we will expand on this general methodology to provide a specific sequence of steps to be followed to generate the desired explanations. The next section describes an implementation of our theory in a cognitive architecture. We will primarily use the following example domain to illustrate the capabilities of the architecture.

#### Example Domain 1 [Robot Assistant (RA)]

Consider a robot that has to find and deliver 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 are characterized by the attributes *size* and *color*. Some other details of the domain include:

- The position of the robot and objects can change due to the execution of one or more actions of the robot.
- The robot can move to a place, pick up or put an object at a particular place, 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 or a part of the object to a particular room or grid cell.

Reasoning occurs over finite time steps with partial knowledge of domain objects and the rules governing the domain dynamics, e.g., the robot knows that objects can only be delivered to people in the same place as the robot; we provide some examples of axioms later in this paper.

We will use a variant of this domain ( $RA^*$ ) to explore the impact of quantization on explanations, e.g., a room with 100 cells instead of four. We also use 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 of this domain 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 the six cells considered on the work table.

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, re-position, bake, etc. each of which can be represented by up to ten fine-resolution actions.

## 4 Reasoning Architecture

Figure 1 shows our overall architecture. It is based on the principle of step-wise refinement and reasons with tightly-coupled transition diagrams at different resolutions. Depending on the domain and tasks, the robot computes and executes plans at two resolutions, but constructs explanations at

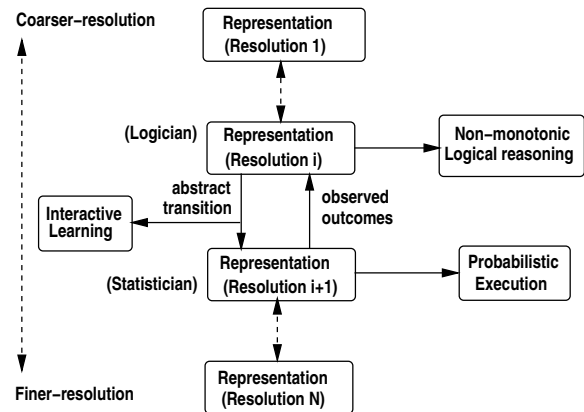


Fig. 1 Architecture represents and reasons with tightly coupled transition diagrams at different resolutions. It combines the complementary strengths of declarative programming and probabilistic reasoning.

other resolutions as needed. For ease of understanding, we focus on two resolutions in the description below, with the fine-resolution transition diagram defined as a *refinement* of the coarse-resolution diagram; we briefly discuss extensions to other resolutions later. 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 automatically *zooming* to and reasoning with the relevant part of the fine-resolution diagram. Each concrete action is executed using probabilistic models of the uncertainty in sensing and actuation, with the relevant outcomes added to the histories at the appropriate resolutions. Reasoning also guides the interactive learning of previously unknown actions, action capabilities, and axioms representing domain dynamics. The architecture combines the complementary strengths of declarative programming, probabilistic reasoning, and relational learning, and is viewed as a logician and statistician working together. Subsets of components, except the theory of explanation and its implementation, are described in other papers (Gomez et al., 2018; Sridharan et al., 2019; Sridharan and Meadows, 2018). We summarize the components here for completeness.

### 4.1 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 Incezan, 2013) to describe the different transition diagrams.  $\mathcal{AL}_d$  has a sorted signature with *statics* (*fluents*), i.e., domain attributes whose truth values cannot (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*.

#### 4.2 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  $\mathcal{H}_c$ .  $\mathcal{D}_c$  comprises a sorted signature  $\Sigma_c$  and axioms governing domain dynamics. For the *RA* domain,  $\Sigma_c$  defines basic sorts such as *place*, *thing*, *robot*, *person*, *object*, and *cup*, arranged hierarchically, e.g., *object* and *robot* are subsorts of *thing*, the sort *step* for temporal reasoning, and instances of sorts, e.g.,  $rob_1$  and  $cup_1$ . For the *RA* domain,  $\Sigma_c$  includes statics such as  $next\_to(place, place)$  and  $obj\_color(object, color)$ , fluents  $loc(thing, place)$  and  $in\_hand(robot, object)$ , and actions  $move(robot, place)$ ,  $give(robot, object, person)$ , and  $pickup(robot, object)$ ; exogenous actions can be included to explain unexpected observations.  $\Sigma_c$  also includes the relation  $holds(fluent, step)$  to imply that a fluent is true at a time step.  $\mathcal{D}_c$  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**  $loc(rob_1, L_1)$ ,  $loc(P, L_2)$

that are used for reasoning. Finally, the history  $\mathcal{H}_c$  of a dynamic domain is typically a record of fluents observed to be true or false at a time step, and the occurrence of an action at a time step. Prior work expanded this notion to represent defaults describing the values of fluents in the initial state. For instance,  $\mathcal{H}_c$  of the *RA* domain encodes “books are usually in the library and if it not there, they are normally in the office”, with the exception “cookbooks are in the kitchen”. For more details, please see (Sridharan et al., 2019).

#### 4.3 Reasoning with Knowledge

Reasoning tasks of a robot associated with a domain description include inference, planning and diagnostics. To do so, the domain description is translated 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*. A literal can thus be true, false or *unknown*. ASP represents recursive definitions and constructs difficult to express in classical logic formalisms, and supports non-monotonic logical reasoning. For coarse-resolution reasoning, program  $\Pi(\mathcal{D}_c, \mathcal{H}_c)$  includes  $\Sigma_c$  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 to let the robot assume the default’s conclusion is false to restore consistency under exceptional circumstances. An *answer set* of  $\Pi$  represents the robot’s beliefs. 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 et al., 2013).

#### 4.4 Refinement, Zooming and Probabilistic Execution

Although reasoning with  $\Pi(\mathcal{D}_c, \mathcal{H}_c)$  provides a plan of actions for any given goal, the robot may not be able to execute some actions or to observe the values of some fluents. For instance, a robot may not be able to directly observe if it is located in a given room, or to pick up an object just because it is in the same room. Actions that cannot be executed directly and fluents that cannot be observed directly are considered to be *abstract*. To implement an abstract transition, we construct a fine-resolution system description  $\mathcal{D}_f$  of transition diagram  $\tau_f$  that is a *refinement* of  $\mathcal{D}_c$ . Refinement may be viewed as looking through a magnifying lens, potentially discovering domain structures that were previously abstracted away (intentionally). We briefly describe the steps below; see (Sridharan et al., 2019) for details.

We first construct a *weak refinement* ignoring the 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 (and actions with magnified sorts) from  $\Sigma_c$ ; and (iv) domain-dependent static relations that relate magnified objects and their counterparts. For the *RA* domain, new basic sorts in  $\Sigma_f$  include:

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

where  $\{c_1, \dots, c_m\}$  are cells in places, *base* and *handle* are components of *cup*, and “ $^*$ ” denotes fine-resolution counterparts. Domain attributes and actions of  $\Sigma_f$  include those of  $\Sigma_c$  modified to reflect the basic sorts of  $\Sigma_f$ :

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

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:

$move^*(R, C)$  **causes**  $loc^*(R, C)$   
 $pickup(R, O)$  **causes**  $in\_hand(R, O)$   
 $pickup^*(R, Cp)$  **causes**  $in\_hand^*(R, Cp)$   
 $loc(O, P)$  **if**  $component(C, P)$ ,  $loc^*(O, C)$

Next, to represent the ability to make observations, our *theory of observation* expands  $\Sigma_f$  to include *knowledge producing* actions that test the value of fluents and changes

*knowledge fluents* describing 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 fluents. For each transition between coarse resolution states  $\sigma_1$  and  $\sigma_2$ , we can show that 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., 2019).

$\mathcal{D}_f$  does not have to be revised unless the domain changes significantly, but reasoning with  $\mathcal{D}_f$  becomes computationally unfeasible for complex domains. For any 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  $\Sigma_f(T)$ . For the transition  $T = \langle \sigma_1, \text{move}(\text{robot}_1, \text{kitchen}), \sigma_2 \rangle$  with  $\text{loc}(\text{robot}_1, \text{office}) \in \sigma_1$  in the *RA* domain,  $\Sigma_f(T)$  includes basic sorts  $\text{robot} = \{\text{robot}_1\}$ ,  $\text{place} = \{\text{office}, \text{kitchen}\}$  and  $\text{place}^* = \{c_i : c_i \in \text{kitchen} \cup \text{office}\}$ , domain attributes  $\text{loc}^*(\text{robot}_1, C)$  taking values from  $\text{place}^*$  and  $\text{loc}(\text{robot}_1, P)$  taking values from  $\text{place}$ , and actions  $\text{move}^*(\text{robot}_1, c_i)$  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., 2019) for details.

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

#### 4.5 Interactive Learning

Reasoning with incomplete knowledge can produce incorrect or suboptimal outcomes. Learning previously unknown actions and axioms may require many labeled examples, which is difficult in robot domains. Also, humans may not have the time and expertise to provide labeled examples or supervision, and an action’s effects may be delayed.

Our architecture includes two schemes for interactively acquiring labeled examples and previously unknown domain knowledge. The first scheme enables active learning of actions and causal laws from human verbal descriptions of the observed behavior of other robots. This scheme assumes that

(a) other robots in the domain (whose behavior can be observed) have the same capabilities as the learner robot; and (b) human description of the observed behavior focuses on one action at a time, and it may be ambiguous but not intentionally incorrect. When human input is available, the learner receives a transcribed verbal description of an action and extracts a relational representation of the observed action’s consequences. Standard natural language processing tools such as a part of speech tagger and the linked synsets of WordNet are used to process the transcribed description to extract sorts, attributes, and actions. The new elements are added to the signature and used with the processed observations to construct new causal laws, incrementally generalizing over time. For instance, processing “the robot is labeling a big textbook” and the observation  $\text{labeled}(\text{book}_1)$  results in the new action  $\text{label}(\text{robot}, \text{book})$  and the causal law  $\text{label}(\text{robot}, \text{book})$  **causes**  $\text{labeled}(\text{book})$ .

The second scheme enables learning of action capabilities and axioms governing domain dynamics, e.g., causal laws and executability conditions. It considers observations obtained either by actively exploring the potential effects of an action, or through (reactive) action execution when an action does not have the expected outcome. This scheme first picks a state transition to be explored further. The task of identifying state-action combinations likely to produce the transition of interest in the presence of immediate or delayed rewards, is posed as a reinforcement learning (RL) problem to mimic interaction with the domain. This basic RL formulation becomes computationally unfeasible for complex domains. To make learning more tractable, we use ASP-based reasoning to automatically restrict learning to object constants, domain attributes, and axioms relevant to the desired transition. To further limit the search space and support generalization, a decision tree is learned based on the relational representation and the examples from RL trials (i.e., states, actions, and rewards experienced). The tree provides a policy to direct exploration in the subsequent RL trials, and candidate axioms that are generalized over time. For more details, see (Sridharan and Meadows, 2018).

#### 4.6 Constructing Explanations

To construct an explanation in response to a query, the robot uses an instantiation of the general methodology described in Section 3.3. Existing software implementations of algorithms enable the robot to determine parts of speech in text (or transcribed verbal input), and select appropriate words and translate their synonym sets into a controlled vocabulary of domain terms (e.g., objects, actions, and relations). Existing software is also used to construct sentences from templates based on the controlled vocabulary, distinguish between physical entities and mental concepts, and to solicit feedback from humans. Although we do not discuss it

in this paper, we also have software that can be used to visually identify domain objects, actions, and spatial relations when we use our architecture on physical robots. The specific steps to be followed are:

1. Parse input query to extract cues (i.e., words and phrases) that match known templates and controlled vocabulary. Given a particular query, e.g., “where is the huge software manual?”, first extract the parts of speech, e.g., *adjective*: ‘big’ and *compound noun*: ‘software manual’, and then identify matching words in the vocabulary, e.g., ‘huge’ = ‘large’ and ‘software manual’ = ‘book’. Also extract key words (e.g., “where”, “why”, “describe”, “detail”) that help determine the kind of explanation to be constructed (more details below).
2. Use cues from query to select a point along the representation abstraction axis, i.e., choose a suitable resolution. Reuse resolution selected for the previous interaction, or use a baseline resolution, unless user query indicates a preference. For instance, if the user input contains the phrase “Please provide a more detailed...”, it directs the robot to select a coarser resolution.
3. Choose points along the communication specificity and verbosity axes using cues extracted from the query. Once again, choose a baseline point or continue with a previous selection unless the query indicates a preference, e.g., the input phrase “Very briefly tell me...” directs the robot to the low end of the verbosity axis.
4. Reason with domain knowledge at the appropriate resolution, and with the identified cues (from query), to compute answer sets (if needed), and to identify *relevant* literals representing knowledge elements (objects, actions, relations). For instance, “what did you do at step 3?” requires the action executed at that time step to be extracted from the answer set, and “why did you move to the *library* at step 2?” requires the robot to compute the answer set before and after the action’s execution, identify changes in beliefs, and to relate these changes to the goal and query.
5. Use chosen points along the three axes, the controlled vocabulary, and the known subject-object-predicate templates, to transform the identified elements to text descriptions. For instance,  $pick\_up(rob_1, book_2)$ , where  $book_2$  is a robotics book, provides the description “the robot picked up the robotics book”. This includes the selection of attributes to use as modifiers, e.g., “a room” or “a medium-sized, library room”, and the choice of the reference symbol, e.g., “a library”, “the library”, or  $study_1$  refer to the same place.

In the specific implementation whose evaluation we report below, we considered two tightly-coupled resolutions (*abstraction* axis). Also, for each requested increase (decrease) in the level of detail, we increased (decreased) by a factor the number of related knowledge elements (*specificity*) and

the level of detail (*verbosity*) used to construct explanations. These choices and the domain’s quantization influence the ambiguity of the explanatory descriptions. High verbosity and high specificity descriptions are unambiguous whereas low verbosity and low specificity descriptions are confusing; also, if rooms have  $10 \times 10$  cells instead of  $2 \times 2$ , the length of the plan and explanation increases. Our software for reasoning and constructing explanations is available in our repository (Meadows and Sridharan, 2018). Note that the methodology and steps for generating explanations are general and can be adapted to other domains, resolutions etc.

## 5 Execution Examples and Results

Our focus in this paper is on exploring how the interplay between knowledge representation, reasoning, and learning supports interactive explanation generation. The coupling between the architecture’s components enables these capabilities but makes it challenging to perform an extensive quantitative evaluation or comparison with other architectures; this is a known problem with cognitive architectures. We have reported the scalability and reliability of knowledge representation, reasoning, and learning with the our baseline refinement-based architecture in other papers (Sridharan and Meadows, 2018; Sridharan et al., 2019). In this section, we thus first discuss execution traces that illustrate the impact of our guiding principles on the ability to construct explanations interactively (Section 5.1). We then measure the reliability and computational cost of generating explanations as a function of the complexity of the domain and the explanations (Section 5.2). We aim to use insights obtained from these traces and experiments to conduct suitable usability studies with human subjects in the future.

### 5.1 Execution Traces

We use execution traces based on the three illustrative domains described in Section 3.3 to examine the failure cases that could exist in the absence of the corresponding guiding principle in the theory. We hypothesize that the robot is able to provide explanations in response to user requests (or queries), and that the approach to provide explanations scales to complex domains and explanations. For ease of understanding, we omit some parts of the explanations and show them as “[...]” in the text.

**Principle #1:** *Explanations should present context-specific information relevant to the domain, task or question under consideration, at an appropriate level of abstraction.*

#### Execution Example 1 [Example 1: 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*<sub>1</sub>. I picked up a large robotics book from *table*<sub>1</sub>. I moved to the *office*. I gave the large robotics book to the *engineer*. I achieved the goal.”
- When asked to provide a more detailed description, the robot revises the point along the abstraction axis; in the current implementation, this corresponds to using the fine-resolution description to generate an explanation.  
**Human:** “Provide a more detailed description.”  
**Robot:** “[...] I moved to cell *c*<sub>2</sub> in the *study*. [...] I moved to cell *c*<sub>4</sub> 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*<sub>2</sub> of the *study*. [...] I moved to cell *c*<sub>7</sub> of the *office*. I gave the large robotics book to the *engineer*. I achieved the goal.”

The absence of this principle would permit explanations lacking in domain information, e.g., the response may be “I moved somewhere. Then I used something to act on something”. The robot may also provide the information at an inappropriate level of abstraction, e.g., the robot may respond to “Briefly tell me what happened” with “I moved to cell *c*<sub>4</sub> in the *library*. I moved to cell *c*<sub>3</sub> in the *library*. I picked up *book*<sub>2</sub>. [...] I moved to cell *c*<sub>6</sub> in the *study* [...]”.

**Principle #2:** *Explanations should be able to provide online descriptions of decisions, rationale for decisions, knowledge, beliefs, experiences that informed the beliefs, and underlying strategies or models.*

#### Execution Example 2 [Example 2: RA Domain]

Consider again the robot in the *RA* domain that has just collected a book from the study and delivered it to the engineer in the office, as in Example 1 above.

- During planning or plan execution, the robot can explain the need for particular actions, and explain their direct and indirect effects.  
**Human:** “Why did you move to cell *c*<sub>4</sub> of the *study*?”  
**Robot:** “I had to retrieve the large robotics book from the table. The table is in cell *c*<sub>4</sub> of the *study*.”
- **Human:** “Relationally describe the effects of moving to cell *c*<sub>4</sub> of the *study*.”  
**Robot:**  $loc^*(rob_1, cell_4), loc(rob_1, study)$ .
- **Human:** “Why do you believe the heavy encyclopedia is in the study?”  
**Robot:** “At time 0, I assumed encyclopedia was in the *library*. At time 4, I observed encyclopedia was in the *study*. To restore consistency, I inferred encyclopedia was in the *study* from time 0.”

Without the second guiding principle, the robot may be incapable of providing suitable explanations for questions about

decisions (“why do you want to pick up [...]?”, “why did you move to [...]?”), beliefs (“why do you believe [...]?”), and related events (“what happened [...]?”, “when was *book*<sub>2</sub> moved to [...]?”). Recent work explains goal changes based on beliefs while planning with incomplete information (Dannenbauer et al., 2018), but this work only presents desiderata without a formal framework to achieve them. Other work proposes a formal theory and uses a belief-desire-intention model to generate reasons that include facts, goals, action outcomes, and failed actions, but the explanations are based on fixed structures and complexity, and their implementation is opaque to natural language (Winikoff et al., 2018). Also, the desired online response in dynamic domains, which is part of this claim, is achieved using the underlying refinement-based framework—see Section 5.2.

**Principle #3:** *Explanation generation systems should have as few task/domain-specific components as possible.*

#### Execution Example 3 [Example 3: RB domain]

Consider a robot that has just completed baking Afghan biscuits and is interacting with a human in the *RB* domain.

- Given the clear separation of concerns between the domain-specific and domain-independent content, our architecture can be adapted to the *RB* domain, although the agent, its task, associated concepts, and the other elements of the signature differ from the *RA* domain.  
**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 to a metallic silver medium baking tray. I picked up the metallic silver medium baking tray. I moved to *table*<sub>2</sub> with the metallic silver medium baking tray in my hand. I opened the silver toaster oven on *table*<sub>2</sub>. I put the metallic silver medium baking tray in the silver toaster oven. I closed the silver toaster oven. I baked the silver metallic medium baking tray in the silver toaster oven for 20 minutes. I opened the silver toaster oven. I picked up the metallic silver medium baking tray from the silver toaster oven. I put the metallic silver medium baking tray on *table*<sub>2</sub>. [...] I achieved my goal.”

The third guiding principle is supported by our architecture’s representational choices, with the content being domain-independent except when the agent interacts with the world (i.e., for sensing or actuation). In the absence of this principle, correct explanations may still be generated for different domains, but a greater effort (human and computational) and architectural changes may be required to represent and reason in different domains. For example, the framework reported by (Rosenthal et al., 2016) is designed for the specific



task of robot navigation; narrating an agent’s behavior in a new domain will require changes to the architecture. Wicaksono et al. (2017) also consider fixed questions and answers for a tool use domain. Roberts et al. (2018) present a theory of plan explanation that is not domain specific, but they make a representational commitment to hierarchical structures and do not instantiate their theory. This claim also ensures that different explanations are provided in different domains for principled reasons and not simply because of the differences between the domains. For example, with our architecture, the difference in quantization between  $RA$  and  $RA^*$  requires no additional programming for the explanation module. Despite a marked increase in the number of steps in the plans for these two domains, there is a significant change in the size of explanations only when asked to explain in a very fine resolution.

**Principle #4:** *Explanation generation systems should consider human understanding and feedback to inform their choices while constructing explanations.*

**Execution Example 4** [Example 4:  $RB$  Domain]

Continue with the previous example in the  $RB$  domain.

- 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<sub>5</sub>* to *mixing\_bowl<sub>1</sub>*. I poured the *cornflakes* in *bowl<sub>2</sub>* to *mixing\_bowl<sub>1</sub>*. I mixed the contents of *mixing\_bowl<sub>1</sub>*. I scraped the mixture in *mixing\_bowl<sub>1</sub>* to *baking\_tray<sub>1</sub>*. I picked up *baking\_tray<sub>1</sub>*. I moved to *table<sub>2</sub>*. I opened *toaster\_oven<sub>1</sub>* on *table<sub>2</sub>*. I put *baking\_tray<sub>1</sub>* in *toaster\_oven<sub>1</sub>*. I baked *baking\_tray<sub>1</sub>* in *toaster\_oven<sub>1</sub>* for 20 minutes. I opened *toaster\_oven<sub>1</sub>*. I picked up *baking\_tray<sub>1</sub>*. I then put *baking\_tray<sub>1</sub>* on *table<sub>2</sub>*. [...] I achieved my goal.”
- **Human:** “Please be more concise.”
 

**Robot:** “I preheated *toaster\_oven<sub>1</sub>*. I mixed *flour*, *butter*, *sugar*, *cocoa*, and *cornflakes* in *mixing\_bowl<sub>1</sub>*. I scraped the mixture to *baking\_tray<sub>1</sub>*. I baked *baking\_tray<sub>1</sub>* in *toaster\_oven<sub>1</sub>* for 20 minutes. I put *baking\_tray<sub>1</sub>* on *table<sub>2</sub>*. 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<sub>2</sub>*?”

**Robot:** “I needed to put the baking tray in the toaster oven that is on *table<sub>2</sub>*.”

**Human:** “How likely is it that there is *cocoa* in the yellow ingredient bowl?”

**Robot:** “I am 95% sure there is no *cocoa* in the yellow ingredient bowl”.

Systems that do not implement this claim may provide explanations that do not match the need of the human user. Without at least an implied theory of mind, the robot can assume that others have made the same inferences given the same knowledge and observations. If human feedback is ignored, it would also become permissible to repeat the same explanation (and not state things more succinctly) when asked to “please be more concise”. Much recent work in explainable AI has considered the task of discerning the mental model(s) a human is operating on (Dannenbauer et al., 2018; Sreedharan et al., 2018).

**Principle #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.*

**Execution Example 5** [Example 5:  $RB$  Domain]

Consider the following exchange in the  $RB$  domain about learned actions and axioms.

- The robot can describe the learned knowledge.
 

**Human:** “What have you learned?”

**Robot:** “I have learned to *chop* an ingredient. I have learned to *spill* an ingredient from a *bowl*.”
- Learned knowledge can be revised over time.
 

**Human:** “What have you learned about action *chop*?”

**Robot:** “It is necessary to hold a *knife* to perform *chop*. It is necessary for the ingredient to be located in front. The action sometimes fails if the ingredient is not *butter*.”
- Knowledge can be acquired from observations obtained from active exploration (e.g., during downtime), reactive action execution (e.g., unexpected and unexplained observation), or by observing other robots.
 

**Human:** “How did you learn that a baking tray can contain ingredients?”

**Robot:** “I poured *cocoa* on a baking tray. I observed baking tray contained *cocoa*.  
or, in a different context:

**Robot:** “I poured *cocoa* into a bowl that was on a baking tray. The observed the bowl did not contain *cocoa*. I observed the baking tray contained *cocoa*.”
- Robot can learn from delayed action outcomes. For example, when a dish set in an oven to be cooked is observed to be burned, the robot infers the reason to be setting the wrong temperature initially.

Our architecture learns new elements to the signature and the axioms—see Section 4.5 and Sridharan and Meadows (2018). Including the new knowledge during reasoning improves the robot’s ability to explain past failure and its own capabilities. If an architecture does not support this guiding principle, the robot’s explanatory power is limited to its initial knowledge. Much of the recent work in explainable AI focuses on plan explanation and does not support

Domain	Low	Medium	High
$RA$	0.00014	0.00025	0.0027
$RA^*$	0.00041	0.0154	0.232

**Table 1** Computation time for domains of different quantization for different points in the space of explanations.

non-monotonic knowledge revision. One counterexample is Wicaksono et al. (2017), which does involve learning action models by selecting and actively exploring an action of interest. However, the approach does not interactively adapt explanations to user needs.

## 5.2 Experimental Evaluation

We evaluated the hypothesis that the ability to construct explanatory descriptions scales to complex, dynamic domains. To do so, we measured the reliability and computational cost of generating explanations as a function of the complexity of the domain and the explanations. The measured computational time did not include planning time or execution time because they are relatively larger and the scalability of planning and execution with a refinement-based architecture has been explored elsewhere (Sridharan et al., 2019).

We conducted 10000 simulated trials in the  $RA$  domain and  $RA^*$  domain for three points of increasing complexity in the space of explanations: (i) “Low” (highest abstraction, lowest specificity, lowest verbosity); (ii) “Medium” (medium abstraction, specificity and verbosity); and (iii) “High” (lowest abstraction, highest specificity, highest verbosity). In each trial, we varied the initial state, goal state, and questions posed to the robot. We also (separately) computed the desired explanations by reasoning with complete knowledge and used these as ground truth (unknown to the robot). The trials were run on a laptop with a 2.40 GHz Intel *i7* CPU. Recall that the  $RA^*$  domain has 25 times as many grid cells in each room as the  $RA$  domain, resulting in many more actions, longer plans (e.g., with  $\approx 40$  steps), and longer explanations. In both domains, a reasonably accurate explanation was obtained in each trial—an explanation is considered to be reasonable if it includes most of the objects and attributes in the ground truth explanation.

Table 1 shows the average results for each quantization and each point in the space of explanations. We observe an increase in the time taken to compute explanations with an increase in the level of quantization. This increase is more pronounced as the complexity of the explanations increases, e.g., the increase in computation time from  $RA$  to  $RA^*$  is more in the “High” column than with “Medium” or “Low”. However, the time taken to compute explanations is not significant in most experimental trials, especially when compared with the planning (or execution) time. Even when asked to provide detailed descriptions in a domain with higher quantization (combination of  $RA^*$  and “High” in Table 1), the robot is able to do so in a reasonable amount of time. Also, it is uncommon to be asked to provide a detailed explana-

tion under a high level of quantization. These results support our hypothesis and indicate the applicability of our architecture to generate explanations in complex, dynamic domains. In other work, we have shown that the underlying architecture for planning with incomplete commonsense knowledge scales to more complex domains. These results thus also indicate the feasibility of introducing more complex models of cognition and learning to generate richer explanations.

## 6 Discussion and Future Work

In this paper, we have formalized the process of providing explanatory descriptions of decisions, beliefs, and experiences in human-robot collaboration. Specifically, we described a theory of explanations comprising (i) claims about representing, reasoning with, and learning knowledge to support explanatory descriptions; (ii) three axes to characterize these descriptions; and (iii) a methodology for constructing these descriptions. We also described an implementation of this theory that is fully integrated with, and strongly influenced by, the representation, reasoning, and learning capabilities of the underlying refinement-based architecture. This architecture uses tightly-coupled transition diagrams at different resolutions to support non-monotonic logical reasoning and probabilistic reasoning with commonsense knowledge and sensor inputs. We also described execution traces and results demonstrating the impacts of our theory on the scalable construction of explanatory descriptions.

Our work opens up multiple directions for further research. First, in this paper, representation and reasoning was limited to two resolutions for ease of explanation. However, other experiments (not reported here) indicate that concepts such as refinement and relevance apply to additional resolutions. Future work will explore the automatic transfer of information and control between multiple resolutions, constructing explanations on demand at the desired level of abstraction. Second, our current architecture does not provide partial explanations, i.e., explanations of some subset of the observations. Future work will explore providing such partial explanations by limiting reasoning to, and choosing the operating point along the three axes, based on the observations of interest. Third, we will use the insights gained from the experiments reported in this paper to conduct studies with human subjects. These studies will evaluate the effectiveness and usability of our theory of explanations and its implementation; the corresponding results will help revise the claims, methodology, and the architecture. Finally, the results reported in this paper were only based on experiments in simulation, although the planning and diagnostics capabilities of the refinement-based architecture have been evaluated on physical robots. In the future, we will evaluate the ability to provide explanatory descriptions on one or more robots sensing and interacting with their surroundings and collaborating with humans in complex domains.

## References

- Anjomshoae S, Najjar A, Calvaresi D, Framling K (2019) Explainable agents and robots: Results from a systematic literature review. In: International Conference on Autonomous Agents and Multiagent Systems, Montreal, Canada
- Balai E, Gelfond M, Zhang Y (2013) Towards Answer Set Programming with Sorts. In: International Conference on Logic Programming and Nonmonotonic Reasoning, Corunna, Spain
- Balduccini M, Gelfond M (2003) Logic Programs with Consistency-Restoring Rules. In: AAAI Spring Symposium on Logical Formalization of Commonsense Reasoning, pp 9–18
- Bohus D, Saw C, Horvitz E (2014) Directions Robot: In-the-wild Experiences and Lessons Learned. In: International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, pp 637–644
- Bollini M, Tellex S, Thompson T, Roy N, Rus D (2013) Interpreting and Executing Recipes with a Cooking Robot. In: J Desai OK G Dudek, Kumar V (eds) Experimental Robotics, Springer Tracts in Advanced Robotics, vol 88, Springer, Heidelberg, pp 481–495
- Borgo R, Cashmore M, Magazzeni D (2018) Towards Providing Explanations for AI Planner Decisions. In: IJCAI Workshop on Explainable Artificial Intelligence, pp 11–17
- Brown R, Kleeck MHV (1989) Enough Said: Three Principles of Explanation. *Journal of Personality and Social Psychology* 57(4):590–604
- Chakraborti T, Sreedharan S, Zhang Y, Kambhampati S (2017) Plan Explanations as Model Reconciliation: Moving Beyond Explanation as Soliloquy. In: International Joint Conference on Artificial Intelligence, pp 156–163
- Dannenbauer D, Floyd M, Magazzeni D, Aha D (2018) Explaining Rebel Behavior in Goal Reasoning Agents. In: ICAPS Workshop on Explainable Planning, pp 12–18
- Dey AK (2009) Explanations in context-aware systems. In: Fourth International Conference on Explanation-Aware Computing, pp 84–93
- Feiner SK, McKeown KR (1989) Coordinating Text and Graphics in Explanation Generation. In: Workshop on Speech and Natural Language, Association for Computational Linguistics, pp 424–433
- Friedman M (1974) Explanation and scientific understanding. *The Journal of Philosophy* 71(1):5–19
- Gelfond M, Incezan 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, Riley H (2018) Representing and Reasoning with Intentional Actions on a Robot. In: ICAPS Planning and Robotics Workshop, Netherlands
- Gregor S, Benbasat I (1999) Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS Quarterly* pp 497–530
- Grice HP (1975) Logic and Conversation. In: Cole P, Morgan J (eds) *Syntax and semantics*, Academic Press, New York, pp 41–58
- Johnson WL (1994a) Agents that explain their own actions. In: Fourth Conference on Computer Generated Forces and Behavioral Representation, pp 87–95
- Johnson WL (1994b) Agents that learn to explain themselves. In: AAAI National Conference on Artificial Intelligence, pp 1257–1263
- de Kleer J, Mackworth AK, Reiter R (1992) Characterizing Diagnoses and Systems. *Artificial Intelligence* 56(2-3):197–222
- Koh PW, Liang P (2017) Understanding Black-box Predictions via Influence Functions. In: International Conference on Machine Learning (ICML), Sydney, Australia, pp 1885–1894
- Langley P, Meadows B, Sridharan M, Choi D (2017) Explainable Agency for Intelligent Autonomous Systems. In: Innovative Applications of Artificial Intelligence (IAAI), San Francisco, USA
- McKeown KR, Swartout WR (1987) Language generation and explanation. *Annual Review of Computer Science* 2(1):401–449
- Meadows B, Sridharan M (2018) Software implementing theory of explanation. <https://github.com/bmeadows/Theory-of-explanations/>
- Miller T (2017) Explanation in artificial intelligence: Insights from the social sciences. Technical report, <https://arxiv.org/abs/1706.07269>
- Reiter R (1987) A Theory of Diagnosis from First Principles. *Artificial Intelligence* 32:57–95
- Ribeiro M, Singh S, Guestrin C (2016) “Why should I trust you?” Explaining the Predictions of any Classifier. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp 1135–1144
- Roberts M, Monteath I, Sheh R, Aha D, Jampathom P, Akins K, Sydow E, Shivashankar V, Sammut C (2018) What was I planning to do? In: ICAPS Workshop on Explainable Planning, pp 58–66
- Rosenthal S, Selvaraj S, Veloso M (2016) Verbalization: Narration of Autonomous Robot Experience. In: International Joint Conference on Artificial Intelligence, pp 862–868
- Sheh R (2017) Different XAI for different HRI. In: AAAI Fall Symposium on Artificial Intelligence for Human-Robot Interaction (Technical Reports), pp 114–117

- Southwick RW (1991) Explaining reasoning: An overview of explanation in knowledge-based systems. *The Knowledge Engineering Review* 6(1):1–19
- Sreedharan S, Madhusoodanan MP, Srivastava S, Kambhampati S (2018) Plan Explanation Through Search in an Abstract Model Space: Extended Results. In: ICAPS Workshop on Explainable Planning, pp 67–75
- Sridharan M, Meadows B (2018) Knowledge Representation and Interactive Learning of Domain Knowledge for Human-Robot Collaboration. *Advances in Cognitive Systems* 7:77–96
- Sridharan M, Meadows B (2019) A Theory of Explanations for Human-Robot Collaboration. In: AAAI Spring Symposium on Story-Enabled Intelligence, Stanford, USA
- Sridharan M, Gelfond M, Zhang S, Wyatt J (2019) REBA: A Refinement-Based Architecture for Knowledge Representation and Reasoning in Robotics. *Journal of Artificial Intelligence Research* 65:87–180
- Thomason J, Zhang S, Mooney RJ, Stone P (2015) Learning to interpret natural language commands through human-robot dialog. In: International Joint Conference on Artificial Intelligence, pp 1923–1929
- Wicaksono H, Sammut C, Sheh R (2017) Towards Explainable Tool Creation by a Robot. In: IJCAI Workshop on Explainable Artificial Intelligence, pp 61–65
- Winikoff M, Dignum V, Dignum F (2018) Why Bad Coffee? Explaining Agent Plans with Valuing. In: ICAPS Workshop on Explainable Planning, pp 36–44
- Zhang Y, Sreedharan S, Kulkarni A, Chakraborti T, Zhuo HH, Kambhampati S (2017) Plan explicability and predictability for robot task planning. In: International Conference on Robotics and Automation, pp 1313–1320