Extended Abstract: Explainable Agency in Ad hoc Collaboration between Humans and Embodied AI*

Hasra Dodampegama[1], Mohan Sridharan[1]

[1]Institute of Perception, Action and Behavior, School of Informatics, University of Edinburgh, UK

Abstract

An assistive AI agent often has to collaborate with previously unseen humans. Methods considered state of the art for such *ad hoc teamwork* use a large labeled dataset of prior observations to model the behavior of other agents and to determine the ad hoc agent's behavior. These approaches are resource-hungry, and do not support rapid incremental revisions or transparency. Our previous work introduced an architecture that enabled an ad hoc agent to choose its actions in simple simulated domains based on non-monotonic logical reasoning with prior domain knowledge and models learned from limited examples to predict the behavior of other agents. Here, we extend this architecture to enable an ad hoc (AI) agent to collaborate with a human performing household tasks in a complex indoor environment, focusing on the ad hoc agent's ability to reason with relevant knowledge and provide relational descriptions as explanations of its behavior and that of the human. We evaluate our architecture's capabilities in *VirtualHome*, a realistic, physics-based, 3D simulation environment.

Keywords

Ad hoc teamwork, Non-monotonic logical reasoning, Ecological rationality, Explainable agency.

1. Motivation

Consider the scenario in Figure 1 from the *VirtualHome* simulation environment, in which an AI agent collaborating with a previously unseen human agent to perform household tasks (e.g., make breakfast, clean dishes). The agent has to reason with different descriptions of prior commonsense domain knowledge and uncertainty. This includes qualitative and metric descriptions of some attributes of the domain and the agent, some rules governing actions and change, and statements that are true in all but a few exceptional circumstances. The agent may have to revise this knowledge over time, and it may have a limited view of the domain and limited communication bandwidth. This scenario is an instance of *Ad Hoc Teamwork (AHT)*, the problem of enabling an agent to cooperate with others without prior coordination [1], which arises in many practical applications.

The state of the art in AHT [2] has evolved from using predetermined policies for specific states [3] to probabilistic or deep network methods that use a history of prior experiences to model the behavior of other agents (or agent types) and optimize the ad hoc agent's behavior [4, 5, 6]. However, it is difficult to gather large datasets of different situations in practical domains. Also, these methods lack transparency, and make it difficult to revise existing knowledge over time. Unlike existing work, our prior work developed a knowledge-guided architecture for AHT, enabling an ad hoc agent to determine its actions based on non-monotonic logical reasoning with prior domain knowledge and rapidly-learned predictive models of other agents' behaviors [7, 8]. In this paper, we describe an extension that focuses on the key functional capabilities of *explainable agency* such as providing on-demand justification of decisions by considering alternative choices, presenting information at a suitable level of abstraction, and communicating information such that it makes contact with human concepts such as beliefs and goals [9]. This architecture draws on cognitive systems research, particularly the benefits of different representations, reasoning schemes, and learning methods [10, 11, 12], enabling an ad hoc agent to:

1. Automatically identify and perform non-monotonic logical reasoning with relevant commonsense domain knowledge and a rapidly-learned predictive model of a human agent; and

XAI-KRKG@ECAI25: First International ECAI Workshop on eXplainable AI, Knowledge Representation and Knowledge Graphs, October 25–30, 2025, Bologna, Italy

*This work is an extended abstract summarizing a paper accepted at 17th International Conference on Logic Programming and Non-monotonic Reasoning October 11-14, 2024 - Dallas, Texas, USA.

hasra.dodampegama@ed.ac.uk (H. Dodampegama); m.sridharan@ed.ac.uk (M. Sridharan)
0000-0003-2302-1501 (H. Dodampegama); 0000-0001-9922-8969 (M. Sridharan)

© 2025 Copyright © 2025 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-4172/short5.pdf

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-02-24

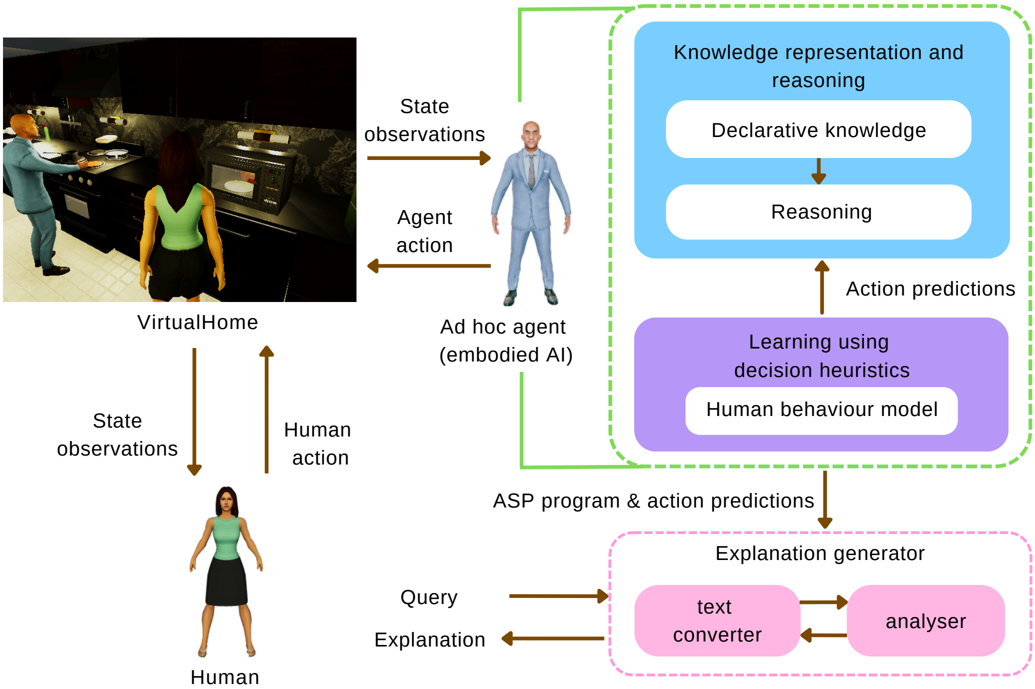

Figure 1: REACT combines strengths of knowledge-based and data-driven methods for reasoning and learning.

2. Automatically construct on-demand relational descriptions of its decisions and those of the human agent as *explanations* in response to different types of questions.

We evaluate these capabilities in the context of household tasks performed in *VirtualHome*, a 3D simulation environment that allows us to explore an agent's interaction with the environment and other agents in a realistic and physically consistent manner [13].

2. Architecture

Figure 1 illustrates our architecture, *Reasoning and Explanations for Ad hoc Collaboration in Teams* (REACT), for human-AI collaboration. REACT's components are described using the following example.

Example Domain 1. *[Example Embodied AI Agent Domain]*
Consider an embodied AI agent and a human agent collaborating to complete household tasks; Figure 1 (top left) shows a snapshot while preparing breakfast. They can interact with the environment through high-level actions, e.g., move to places, pick up or place objects, switch appliances on or off, and open or close appliances. Completing a task requires a sequence of such actions to be executed by the AI agent or the human who do not communicate directly with each other. Prior commonsense knowledge of the AI agent includes relational descriptions of some attributes of the domain, itself, and the human (Section 2.1); a learned (or encoded) graph of information about likely locations of objects in the domain; default statements that hold in all but a few exceptional circumstances; and some axioms governing actions and change, e.g., the agent can only pick one object at a time. The agent assumes that the human has access to the same state information and will make (what it considers) rational decisions. It determines its action choices through non-monotonic logical reasoning with the prior domain knowledge and an incrementally learned model of the human's behavior, as described below.

2.1. Knowledge Representation and Reasoning

In REACT, the transition diagram of any given domain is described using an extension of action language $\mathscr{AL}_d$ [14]. The domain representation comprises a system description $\mathscr{D}$, a collection of statements of $\mathscr{AL}_d$, and a history $\mathscr{H}$. $\mathscr{D}$ has a sorted signature Σ with basic sorts such as *object*,

$appliances$, *ad_hoc_agent*, *human*, and $step$ (for temporal reasoning) in our example domain; actions such as $grab(ad\_hoc\_agent, object)$ and $switch\_on(ad\_hoc\_agent, appliances)$; statics, i.e., domain attributes whose values cannot be changed by actions; and fluents, i.e., attributes whose values can be changed by actions. Basic sorts are arranged hierarchically, e.g., *microwave* is a sub-sort of *electricals* that is a sub-sort of $appliances$, a sub-sort of *object*; and action $find$ includes *move* and *rotate* actions. Fluents can be *inertial*, i.e., they obey laws of inertia and are changed by actions, e.g., $in\_hand(ad\_hoc\_agent, object)$ describes an object being held by the ad hoc agent; and *defined*, i.e., they do not obey inertia laws and are not directly changed by the agent's actions, e.g., $exo\_hand(human, object)$ describes the human holding an object. Based on Σ, the domain dynamics are described in $\mathscr{D}$ using axioms such as:

$$grab(A, O) \textbf{ causes } in\_hand(A, O) \tag{1a}$$

$$heated(F) \textbf{ if } on(F, E), switched\_on(E), food(F) \tag{1b}$$

$$\textbf{impossible } grab(A, O) \textbf{ if } on(O, E), not\ opened(E) \tag{1c}$$

Statement 1(a), a causal law, implies that grabbing an object causes it to be in the hand of the ad hoc agent; Statement 1(b), a state constraint, implies that a food item placed in an electrical appliance (e.g., microwave) that is switched on gets heated; and Statement 1(c), an executability condition, prevents the ad hoc agent from trying to grab an object from an appliance with a closed door. History $\mathscr{H}$ is a record of observations of the form $obs(fluent, boolean, step)$, and action executions of the form $hpd(action, step)$ at specific time steps. It also includes default statements that are true in the initial state.

To reason with knowledge, we automatically construct program $\Pi(\mathscr{D}, \mathscr{H})$ in CR-Prolog, an extension to ASP that supports consistency restoring (CR) rules. $\Pi(\mathscr{D}, \mathscr{H})$ includes statements from $\mathscr{D}$ and $\mathscr{H}$, inertia axioms, reality check axioms, closed world assumptions for defined fluents and actions, helper relations, e.g., $holds(fluent, step)$ and $occurs(action, step)$ to imply that a fluent is true and an action is part of a plan at a time step, and helper axioms that define goals and guide planning and diagnostics. ASP encodes *default negation* and *epistemic disjunction*, and supports non-monotonic logical reasoning, an essential ability for agents reasoning and acting based on incomplete knowledge and noisy observations. The CR rules allow the agent to make assumptions (e.g., that a default statement does not hold) under exceptional circumstances to recover from inconsistencies. All reasoning tasks (i.e., planning, diagnostics, and inference) are then reduced to computing *answer sets* of Π. We use the SPARC system [15] to solve CR-Prolog programs. Example programs are in our open source repository [16].

Our example scenario is complex, with many objects and actions; the corresponding tasks require multi-step plans. To enable the efficient use of logical reasoning in such scenarios, REACT represents and reasons with knowledge at two (formally-coupled) levels of abstraction by building on our prior work on a refinement-based architecture [17]. Also, as the ad hoc agent traverses through our example environment or other similar environments, it collects statistics of relevant locations of objects and human action preferences, using this information to automatically restrict grounding and simplify reasoning, e.g., depending on goal and specific actions in the plan, the ad hoc agent can automatically select the relevant signature and restrict axioms to this signature.

2.2. Agent Behavior Models

REACT enables the ad hoc agent to reason with models that predict the human agent's action(s) in any given state. Similar to our prior work, REACT uses the *Ecological Rationality* (ER) principle [18], which is based on Herb Simon's definition of *Bounded Rationality*, and an algorithmic theory of heuristics, enabling the ad hoc agent to rapidly learn and revise these predictive models from limited data.

Specifically, REACT enables the ad hoc agent to learn an ensemble of *Fast and Frugal* (FF) trees that predict the human agent's behavior; each FF tree provides a binary choice for a particular action and the number of leaves in the tree is limited by the number of attributes [19]. An example of one such FF tree in an ensemble is in Figure 2. The initial version of these trees were built using only 100 traces of human agent's action choices and domain states from the VirtualHome domain, with the corresponding attributes in Table 1. Furthermore, consistent agreement (disagreement) between observations and

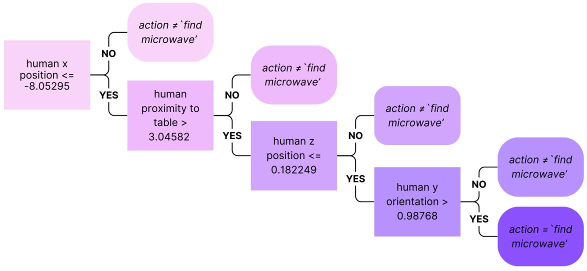

Figure 2: FF tree for *find(microwave)* action.

Table 1: Attributes used to create behavior model.

Description of the attribute
Two previous actions of the human
Two previous objects human interacted with
Position and orientation of the human
Distance from human to kitchen table
Distance from human to kitchen counter
Number of objects on kitchen table

predictions of an existing model triggers model choice (revision); the ad hoc agent is thus able to quickly adapt to changes in the domain or the human's behavior.

2.3. Transparency in decision making

An automated decision-making system's ability to justify its choices promotes acceptability [20, 21]. Unlike methods that seek to make a complex learned model interpretable, or to explain all the choices made by a reasoning system, REACT responds to any given question about specific decisions by quickly identifying the relevant information and constructing relational descriptions. It does so by building on the underlying logic-based representation of knowledge and simple predictive models, and on prior work in our group that identified the axioms and literals relevant to the questions posed to an AI system making decisions [22]. Specifically, REACT enables the ad hoc agent to respond to four types of questions identified as being important in work on explainable planning [21].

1. **(Causal questions)** *Why did you execute a_I, i.e., action a at step I?*
   - If a_I is not the last action of plan P executed by the agent, extract actions $\{A_{af}\} \in P$ that occurred after a_I. Identify axioms that have negation of $a_{I+i} \in \{A_{af}\}$ in *head*, i.e., prevent a_{I+i} from occurring.
   - Check if each such axiom's *body* is satisfied by *answer set* at I. If yes, identify fluent literals f in *body* whose value changed from I to $I+1$. Collect all such literals $\{f\}$ over all identified axioms have been changed by the execution of a_I. Use literals to construct answer.
   - If a_I is last action in P, it contributed to achieving the goal. Use goal to construct answer.
2. **(Contrastive questions)** *Why did you not execute a_I, i.e., action a at I?*
   - Identify axioms with the negation of a_I in *head*. If any such axiom's *body* is satisfied by *answer set* at I, collect fluent literals $\{f\}$ in *body* as they prevented consideration of a_I.
   - If not, identify axioms in $\Pi(\mathscr{D}, \mathscr{H})$ with a_I in its *body* with other literals $\{f'\}$, i.e., causal laws. Extract the precondition literals $\{f'\}$ and identify axioms with $l \in \{f'\}$ in its *head*, i.e., state constraints. Check if each such axiom's *body* is satisfied by *answer set* at step I. If not, use literal $l \in \{f'\}$ and the *body* literals of axiom to construct answer.
3. **(Justify beliefs)** *Why did you believe l_I, i.e., l at step I?*
   - Replace ground terms of l_I with variables. Identify axioms with l_I in *head*, and check whether *body* is satisfied by answer set at step I. If yes, collect fluent literals $\{f\}$ in *body* as they support l_I.
   - Create a tree with l_I as its head and each selected axiom as a branch. With each axiom, store the supporting $\{f\}$ fluents. Repeat for each fluent literal in $\{f\}$ as target belief until no more axioms are identified. Use collected literals for answer.
4. **(Counterfactual Questions)** *What will be the outcome of executing a'_I?*
   - Retrieve most recent state of environment S_{I-n} with respect to step I. Retrieve corresponding action for ad hoc agent (a_{I-n}) and human ($\{a^{ot}_{I-n}\}$, use predictive models).
   - Perform mental simulation of the future step $I-n+1$ from S_{I-n} using existing knowledge and action choices of agents $\{a^{ot}_{I-n}, a_{I-n}\}$. Repeat n times, i.e., roll out the future and explore effects of actions of ad hoc agent and human until queried step I. Collect state information S_I at I.
   - Retrieve agent's action a_I and predicted human action a^o_I in S_I. Traverse FF tree model of human to identify and collect active branches for a^o_I to construct answer.

- Replace action a_I^o (a_I) with a_I' for the human (ad hoc agent). Roll out environment one step to obtain the resultant state S_{I+1} to be used to construct answer.

For each type of question, identified literals are processed using existing tools and templates to generate descriptions as responses (i.e., explanations) before, during, or after planning or execution.

3. Experimental Setup and Results

We experimentally evaluated two hypotheses regarding REACT's capabilities:

- **H1:** The combination of reasoning and learning in REACT enables the ad hoc agent to adapt to changes and perform better than a logic-based baseline; and
- **H2:** REACT enables the ad hoc agent to generate relational descriptions as explanations of its decisions and beliefs and those of the human.

The performance measures were precision and recall of explanations, number of steps (i.e., plan length), and total time taken (by human and AI agent) to complete the task.

3.1. Experimental Setup

In the VirtualHome domain, we modeled the human as a simulated entity that chooses its actions based on an ASP program. The human was assigned the same goal as the embodied AI (ad hoc) agent, e.g., *prepare and eat breakfast.* The **baseline operation** involved each of them receiving the same observations of the domain at each step; they then continued with their planned actions or computed a new plan as needed. There was no direct communication between them, and they did not have any prior knowledge or model of each other's capabilities. With REACT, the key difference was that the ad hoc agent used the learned models to predict a couple of (future) actions of the human, and its ASP program included additional axioms for reasoning about these predicted actions. *The agent's plan thus expects the preconditions for some intermediate steps to be created by the actions executed by the human although the human may not always do so.* The human, on the other hand, could not predict the ad hoc agent's actions and the axioms to reason about these actions were not included in the corresponding ASP program. The human's actions were primarily determined by the current state and goal. The human's ASP program also includes actions such as eating and drinking that were not available to the ad hoc agent. Furthermore, the human's ASP program encoded some priorities and preferences, e.g., when preparing breakfast, the human toasted the bread first before preparing the cereal.

As described in Section 2.2, each model predicting human behavior was an ensemble of FF trees based on just 100 prior traces of human actions and domain state. These observations also provided priors regarding likely location of objects, which were used to simplify planning (end of Section 2.1).

To evaluate **H1**, we designed **Exp1** by first constructing 720 different configurations, each with a different arrangement of objects in the initial condition, e.g., bread on the kitchen table instead of the counter, microwave open instead of closed. We then randomly chose 100 of these configurations and measured the ability to achieve the shared goal (e.g., prepare and eat breakfast) with each of our options: REACT and baseline. To evaluate **H2**, we designed **Exp2** in which we randomly selected 10 configurations (from the 100 in **Exp1** and saved the corresponding answer sets (with REACT, baseline) to provide ground truth. Then, we posed 32 different questions (divided between four types of questions) about some chosen steps in each trial corresponding to one of these 10 configurations, with answers computed as described in Section 2.3. We recorded the precision and recall of retrieving literals to answer these questions. Furthermore, we considered execution traces as qualitative evaluation of **H2**.

3.2. Experiment Results

Table 2 summarizes the results of **Exp1**. Since the plan length and task completion time vary substantially based on the initial configuration, we computed the values of the performance measures for REACT as a fraction of those for the baseline in each trial, and reported the average of these ratios in Table 2. We observed that REACT significantly reduces the number of steps and the time taken to achieve the goal

Table 2
Number of steps and time taken by REACT to achieve the goal, expressed as a fraction of these values for the baseline. REACT substantially improves performance.

Architecture	Steps	Time
REACT	0.89 ± 0.11	0.90 ± 0.19
Baseline	1 ± 0.05	1 ± 0.04

Table 3
Explanation generation: REACT provides high precision and recall of retrieving relevant literals.

Question type	*Precision*	*Recall*
Action justification	1.00	1.00
Contrastive	0.89	0.99
Belief justification	0.88	0.94
Counterfactual	1.00	0.78

compared with the baseline, supporting **H1**. The prediction accuracy of the behavior models learned by the ad hoc agent for the human (Section 2.2) was **85.19**%, i.e., the model does make errors, but it supports rapid revision and reasoning with it and prior knowledge significantly improves performance. In other experiments (not reported here), we noticed that the ad hoc agent was able to revise these models to track (or adapt to) changes in the human's behavior.

Table 3 summarizes results of **Exp2**, with precision and recall calculated relative to all the literals expected to be included in the explanation. The observed high values of precision and recall indicate the ability to automatically extract the correct literals to provide relational descriptions as explanations in response to different types of questions, thus supporting **H2**. As an additional example, consider an exchange with the ad hoc agent after it collaborated with the human to prepare breakfast.

- **Questioner:** "Why did you find bread slice in step 0 ?"
- **Ad hoc Agent:** "Because I had not found the bread slice yet and wanted to grab it in step 1". This response identifies an action's outcome being a requirement for a subsequent action.
- **Questioner:** "Why did you not find the water glass in step 0?"
- **Ad hoc Agent:** "Because I predicted that the human will find the water glass in step 0." The agent may be asked about the human's actual or hypothetical action choices.
- **Questioner:** "Why do you think human will grab water glass in step 1?"
- **Ad hoc Agent:** "Because my prediction is that the human wants to bring the glass to the table."
- **Questioner:** "What if the human decided to grab the cutlets in step 1 ?"
- **Ad hoc Agent:** "If the human grabs cutlets in step 1, they will be in human's hands in step 2." To answer questions about hypothetical situations, the ad hoc agent simulates the evolution of state, and the execution of actions by the human and the agent.

Additional videos are available in our code repository [16].

4. Conclusions

This paper described REACT, an AHT architecture for an embodied AI agent to collaborate with a human by reasoning with prior commonsense domain knowledge and incrementally learned models predicting the behavior of the human. REACT combines the principles of non-monotonic logical reasoning and ecological rationality, automatically identifying and reasoning efficiently with the relevant knowledge and observations. Also, the interplay between reasoning and learning enables the embodied AI agent to provide relational descriptions as on-demand explanations of its own decisions and those of the human. Experimental evaluation in a physics-based simulation environment demonstrates performance improvement compared with a logical reasoning baseline. Future work will explore: (a) human-controlled avatars in our current experiments; (b) incremental learning of domain knowledge in more complex domains; (c) scalability to multiple ad hoc agents and humans; and (d) implementation and evaluation on physical robots in AHT settings.

Acknowledgments

This work was supported in part by the U.S. Office of Naval Research Award N00014-20-1-2390. All conclusions are those of the authors alone.

Declaration on Generative AI

Generative AI tools have not been used in writing this paper.

References

[1] P. Stone, G. Kaminka, S. Kraus, J. Rosenschein, Ad Hoc Autonomous Agent Teams: Collaboration without Pre-Coordination, in: AAAI Conference on Artificial Intelligence, 2010, pp. 1504–1509.

[2] R. Mirsky, I. Carlucho, A. Rahman, E. Fosong, W. Macke, M. Sridharan, P. Stone, S. Albrecht, A Survey of Ad Hoc Teamwork: Definitions, Methods, and Open Problems, in: European Conference on Multiagent Systems, 2022.

[3] M. Bowling, P. McCracken, Coordination and adaptation in impromptu teams, in: National Conference on Artificial Intelligence, 2005, p. 53–58.

[4] S. Barrett, P. Stone, S. Kraus, A. Rosenfeld, Teamwork with limited knowledge of teammates, in: AAAI Conference on Artificial Intelligence, volume 27, 2013.

[5] M. A. Rahman, N. Hopner, F. Christianos, S. V. Albrecht, Towards open ad hoc teamwork using graph-based policy learning, in: International Conference on Machine Learning, 2021.

[6] J. Zand, J. Parker-Holder, S. J. Roberts, On-the-fly strategy adaptation for ad-hoc agent coordination, in: International Conference on Autonomous Agents and Multiagent Systems, 2022, p. 1771–1773.

[7] H. Dodampegama, M. Sridharan, Back to the Future: Toward a Hybrid Architecture for Ad Hoc Teamwork, in: AAAI Conference on Artificial Intelligence, 2023.

[8] H. Dodampegama, M. Sridharan, Knowledge-based Reasoning and Learning under Partial Observability in Ad Hoc Teamwork, Theory and Practice of Logic Programming 23 (2023) 696–714.

[9] P. Langley, B. Meadows, M. Sridharan, D. Choi, Explainable Agency for Intelligent Autonomous Systems, in: Innovative Applications of Artificial Intelligence, San Francisco, USA, 2017.

[10] J. E. Laird, K. Gluck, J. Anderson, K. D. Forbus, O. C. Jenkins, C. Lebiere, D. Salvucci, M. Scheutz, A. Thomaz, G. Trafton, R. E. Wray, S. Mohan, J. R. Kirk, Interactive Task Learning, IEEE Intelligent Systems 32 (2017) 6–21.

[11] M. Sridharan, Back to the Future of Integrated Robot Systems, in: AAAI Conference on Artificial Intelligence, Philadelphia, US, 2025.

[12] P. H. Winston, D. Holmes, The Genesis Enterprise: Taking Artificial Intelligence to Another Level via a Computational Account of Human Story Understanding, Computational Models of Human Intelligence Report 1, Massachusetts Institute of Technology, 2018.

[13] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, A. Torralba, Virtualhome: Simulating household activities via programs, in: International Conference on Computer Vision and Pattern Recognition, 2018, pp. 8494–8502.

[14] M. Gelfond, D. Inclezan, Some Properties of System Descriptions of AL_d, Applied Non-Classical Logics, Special Issue on Equilibrium Logic and ASP 23 (2013).

[15] E. Balai, M. Gelfond, Y. Zhang, Towards Answer Set Programming with Sorts, in: Conference on Logic Programming and Nonmonotonic Reasoning, 2013.

[16] H. Dodampegama, M. Sridharan, https://github.com/hharithaki/REACT, 2024.

[17] M. Sridharan, M. Gelfond, S. Zhang, J. Wyatt, REBA: A Refinement-Based Architecture for Knowledge Representation and Reasoning in Robotics, Journal of Artificial Intelligence Research 65 (2019) 87–180.

[18] G. Gigerenzer, What is Bounded Rationality?, in: Routledge Handbook of Bounded Rationality, Routledge, 2020.

[19] K. Katsikopoulos, O. Simsek, M. Buckmann, G. Gigerenzer, Classification in the Wild: The Science and Art of Transparent Decision Making, MIT Press, 2021.
[20] S. Anjomshoae, A. Najjar, D. Calvaresi, K. Framling, Explainable agents and robots: Results from a systematic literature review, in: International Conference on Autonomous Agents and Multiagent Systems, Montreal, Canada, 2019.
[21] M. Fox, D. Long, D. Magazzeni, Explainable Planning, in: IJCAI Workshop on Explainable AI, 2017.
[22] M. Sridharan, Integrated Knowledge-based Reasoning and Data-driven Learning for Explainable Agency in Robotics, in: Explainable Agency in Artificial Intelligence: Research and Practice, CRC Press, 2024.