



Article

Toward Competent Robot Apprentices: Enabling Proactive Troubleshooting in Collaborative Robots

Christopher Thierauf ^{*}, Theresa Law [†], Tyler Frasca [†] and Matthias Scheutz 

Human-Robot Interaction Lab, Tufts University, Medford, MA 02155, USA; matthias.scheutz@tufts.edu (M.S.)

^{*} Correspondence: christopher.thierauf@tufts.edu[†] Work done while at Human-Robot Interaction Lab, Tufts University.

Abstract: For robots to become effective apprentices and collaborators, they must exhibit some level of autonomy, for example, recognizing failures and identifying ways to address them with the aid of their human teammates. In this systems paper, we present an integrated cognitive robotic architecture for a “robot apprentice” that is capable of assessing its own performance, identifying task execution failures, communicating them to humans, and resolving them, if possible. We demonstrate the capabilities of our proposed architecture with a series of demonstrations and confirm with an online user study that people prefer our robot apprentice compared to robots without those capabilities.

Keywords: fault detection and recovery; proactive communication; robot problem solving

1. Introduction

Our ability to perform collaborative tasks relies on an awareness of our own and our collaborators’ abilities and limitations. Knowing these abilities and limitations allows us to make more effective requests, adapt to failures, modify behaviors, or generally avoid actions that will certainly fail. This is referred to as the “social mind” [1] and has been shown to provide a valuable set of problem-solving tools [2]. In this paper, we work towards providing robots with these capabilities in an effort to develop more autonomous and more effective robot collaborators.

As a motivating example, consider Figure 1. A robot can be controlled by a human using language (e.g., [3–5]). In these approaches, however, the robot can be viewed more as a tool—language is used to specify a series of behaviors to be executed. In contrast, we consider cases where unexpected events may occur and so the robot must become an *apprentice*. A robot can, for example, observe that a grasping task has failed, but will require to work collaboratively with the human expert (who may observe the cause of the failure, or know other task requirements) to arrive at an acceptable solution.

Recent work in robot self-assessments based on past task performance (e.g., [6]) is highly relevant in this regard as past experience can be used to make predictions about future behavior outcomes. But just predicting failures does not address how to overcome them or how to work with a human partner to address ongoing faults. To properly cope with these failures, the robot must be able to understand their impact. It can then use its performance self-assessment to determine the best strategy for completing the task and inform the human about it if the course of actions deviates from what was originally instructed or expected.

In this systems paper, we present an integrated system that makes use of autonomous robot self-assessments integrated into a dialogue interface, goal management system, and action execution system. This produces an agent that is capable of introspecting into expected task performance and using this information for detecting, classifying, and coping with unexpected failures. The system is also capable of engaging in dialogues with human interactants about all aspects of this process, allowing it to reject instructed actions if they are certain to fail, and to propose alternative solutions that have a chance of succeeding.



Citation: Thierauf, C.; Law, T.; Frasca, T.; Scheutz, M. Toward Competent Robot Apprentices: Enabling Proactive Troubleshooting in Collaborative Robots. *Machines* **2024**, *12*, 73. <https://doi.org/10.3390/machines12010073>

Academic Editor: S. M. Mizanoor Rahman

Received: 14 December 2023

Revised: 8 January 2024

Accepted: 14 January 2024

Published: 18 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

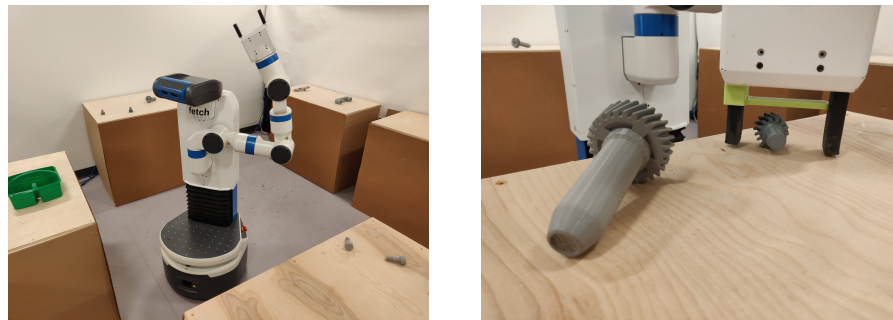


Figure 1. Our Fetch in the FetchIt! environment (**left**) will be forced to address problems such as ambiguous instruction with differing outcomes or an intentionally jammed gripper (both **right**), as well as power failure and jam which restrict action only in specific contexts (not pictured).

To this end, our contribution is a system which, through a tight integration of a cognitive architecture and novel fault detection/communication strategies, is capable of identifying a failure and inferring its impact given past experiences, which it can then communicate to a human partner or handle proactively. We additionally contribute a series of demonstrations of our approach and a human subject study providing evidence supporting this proactive approach.

To present our approach, we discuss previous work, framed in the context of a broader assembled system. We then provide a general implementation strategy in the form of an architecture-level overview of our technical approach, as well as the specific technical contributions that enable some of the key features (self-assessment, dialogue, and failure detection/classification for use in addressing a cost-optimized planning problem). Finally, we demonstrate the capabilities of this system through case studies and human subject study.

2. Background

2.1. Robot Self-Assessment

Robot self-assessment enables robots to predict potential outcomes before, during, and after task execution, providing insight into success or failure probabilities. Examples have largely been constrained within a tightly controlled domain, such as operator scheduling, which aims to enable an agent to be as autonomous as possible while also identifying when (and how) a human partner should take over, if necessary. This has been studied largely with UAVs (unmanned aerial vehicles) [7,8] and other deployed autonomous systems (such as the distributed sensing systems of [9] or the search-and-rescue platforms of [10]). Multi-agent approaches have also been explored [11], some of which utilize game theory models [12], while others use a “neglect”-based model [8,13]. Task-oriented approaches include [14,15], which focus on grasping and handovers, respectively, or [16,17] which focus on task scheduling and assessment within a human-specified task. These systems could likely be adapted to a broader array of problems than their authors have presented, but this remains unexplored.

Like these, our agent works within the confines of its instructions. However, our agent makes continuous assessments of past experiences to make the statistical determination that the robot performance is still within expectation, or what past experience should be referenced if it is not. For example, given the robot’s current inability to grab objects, perhaps it should be making use of a past pool of experience from when its gripper was jammed. This past experience is necessary for setting a baseline for expectation, which is useful for detecting unexpected events and for predicting the success of future actions. Further, it enables inferences beyond what it can immediately observe. As we show in Section 3.5, being able to select the appropriate pool of past experience enables the agent to make use of data that has not been directly observed during the current deployment time.

In our approach, the agent will attempt to observe that the distribution of action outcomes is statistically not within expectation, implying that there is some unexpected and unobserved error state. Systems which attempt to explore novelties have been discussed in

previous work. For example, consider [18], which uses an “artificial curiosity mechanism” to explore unexpected features in the agent’s environment. Closer to our work is [19] in that the system works to infer some hidden state, though we are more interested in using this inference to inform dialogues which explore solutions. In contrast to these systems which explore their environment, ours passively monitors the environment; constructing a model of the environment which can be statistically evaluated (similarly to [20]). However, unlike [20], we do not rely upon a neural model. We instead use a symbolic approach to allow us to explicitly reason about failure events and the impact of these failures to construct long-term plans and to generate dialogue. Purely sub-symbolic approaches, including neural nets, but also including deep learning and other approaches, struggle with these tasks, motivating our use of a symbolic approach.

2.2. Action Selection with Human Interactions

The technical developments behind anticipatory planning (sometimes also called human-aware planning) are not immediately relevant to our work. However, research in this space has explored how robots which directly anticipate human needs are perceived. In particular, this research has explored the relationship between robot decision-making and human–robot trust, team cohesion, and task completion, providing a strong theoretical justification for our interest in our approach.

Past work has demonstrated that systems which anticipate human goals are capable of effective motion and task planning [21,22], as well as cost-optimized planning (such as optimizing for time [23]). Some of these systems operate by maintaining a reasonable model of the human partner’s cognitive state to anticipate human behavior before selecting an action (e.g., [24]). This approach can also be used to determine what information should be communicated when performing a task [25], or to avoid needing communication to the extent that this is possible (e.g., through inferring goals [26] or inferring when a question can be asked [27]).

These anticipatory systems have been shown to be effective for improving the efficiency of human–robot teams. For example, [28,29] demonstrated that a robot system capable of selecting actions, as informed by expected human action, leads to a “dramatic improvement” [29] in team cohesion and task completion, among other effects. Further, these systems have demonstrated their ability to integrate with dialogue systems to both interpret human commands and to follow up with questions [30]. This previous literature demonstrates the value of systems which prioritize human perceptions when task planning, and highlights our interest in this work.

One strategy for determining the appropriate action to select is to construct a cost-optimized planning problem. Cost-optimized planning allows for planning problems to be constructed which account for some action cost; behaviors which can be selected by the agent are associated with some value, and the cumulative value of a plan is maximized or minimized to select the best option. Approaches for assigning the action cost vary widely, with approaches ranging from assigning the action cost based on action time-to-perform [31,32], human-specified costs [33], deviance from human expectation [34], social norms [35], social welfare maximization [36], energy cost [37], and many more beyond our scope.

These costs can be learned from real-time robot experience, which is a realization we will make use of. This is seen in [38] and [6], in which the robot learns success likelihoods through robot experience. In [38], this knowledge is used for a later planning problem, where the agent attempts to maximize success and efficiency. In [6], this is used for human–robot dialogues about potential action success, failures, and counterfactuals. Our system will provide both of these features.

2.3. Task and Performance Dialogues

In [39], the authors present a system that enables robot self-assessments through dialogue. Of particular interest to us is the ability of the robot R to describe action out-

comes through probabilities which are learned through experience of the human H , as illustrated below:

H: Describe how to dance.

R: To dance, I raise my arms, I lower my arms, I look left, I look right, I look forward, I raise my arms, and I lower my arms.

H: What is the probability that you can dance?

R: The probability that I can dance is 0.9.

The robot has a complex action (“dance”) which is known to contain a series of other actions. Through knowledge of the success rate of each of these actions, it can calculate the cumulative probability.

As further taxonomized and discussed in [40], these tight integrations between dialogue and robot self-assessment systems enable an improved human understanding of the robots proficiency at a given task. The question of *what* should be said, as well as when, is non-trivial [41]. In [42], a formalism is proposed to balance between taking time to complete an action versus taking time to communicate with a partner. In both [43,44], communication is represented within the action and planning pipelines of the agent to enable fluent human–robot partner interactions. This fluent interaction is increasingly recognized as an important step in developing effective human–robot interactions [45].

Many human–robot dialogue approaches are human-driven, like ours. It is the human operator who initiates an interaction and drives the conversation, with the robot supplying responses, taking actions, and providing feedback (although we will show that the agent taking some degree of initiative is desirable and preferred). This approach is used in, for example, [46], where dialogues are used in collaborative tasks, or in [47], where agent exploration can provide information for a later dialogue. These approaches of being actively engaged and embodied may have dramatically positive impacts on the performance of the human–robot team (as discussed in more detail in [48,49]). We will, therefore, later demonstrate that our approach contributes to effective human–robot teaming (despite it not being in quite the same category of other robots that are exclusively “tools”).

However, some robot performance dialogue work has been explicitly robot-driven. Consider, for example, [50], where value-of-information theory is used to determine when a robot should query an operator (and with what questions). These approaches contribute value in their ability to engage in dialogues which are triggered by some observation, and we will make use of this approach—when a problem with the human partner’s plan is observed, it is the responsibility of the agent to detect, interject, and solve.

3. Technical Approach

3.1. Environment and Tasks

We will consider scenarios where a Fetch robot R receives an instruction from a human H to perform an assembly task that requires R to gather several objects and place them in the “caddy” item, a subset of the ICRA 2019 FetchIt! challenge [51] in which the Fetch mobile manipulator [52] must autonomously navigate an enclosed area to detect, grasp, and retrieve a set of objects. To successfully complete its task, the robot has a set of perception, manipulation, and navigation actions. As described in [53], these actions can be assembled into sequences to create larger “action scripts”. Available actions include `approach(location)`, `grab(object)`, `fetch(object)`, and others. The agent is already aware of some task-relevant objects (the caddy, screw, small and large gear, gearbox top and bottom) as well as their location on each table. For basic robot behaviors, we utilize the manufacturer-provided software stack.

We modify the task in different ways to induce failures, e.g., by intentionally preventing the robot’s grippers from closing using a small plastic component wrapped around one of the gripper fingers (this device is visible in Figure 1, right). The agent is not aware of the potential failures (in this case, the gripper cannot grasp), and instead begins with the assumption that it is fully operational. It must appropriately select actions and come to its own conclusions about its ability to solve tasks autonomously when possible.

To enable this behavior, we take a systems-level approach to assembling a cognitive architecture which allows the agent to perceive its environment and plan actions towards various goals. We extend goal management and logical reasoning systems that enable task planning and execution by enabling them to assess failure likelihoods, emulating “what if” reasoning: Is the task likely to succeed as is, or what if a new plan were substituted? Our architecture is presented as Figure 2, and discussed in more detail next.

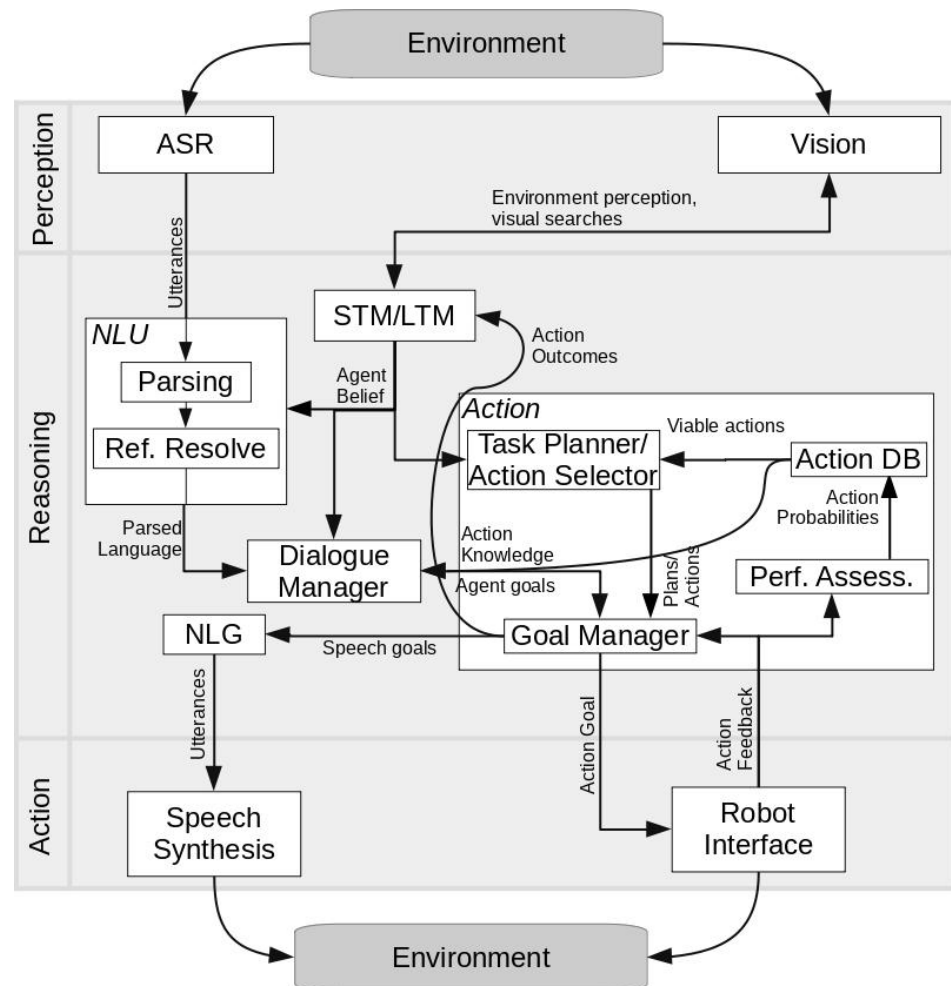


Figure 2. The robot architecture integrating the methods from [54] for self-assessment. Abbreviations: automatic speech recognizer (ASR); short-term memory (STM); long-term memory (LTM); reference (Ref.); performance (Perf.)

3.2. Sensing

Our agent employs two sensing modalities: visual sensing and auditory sensing, the former to obtain symbolic representations of objects in the environment (utilizing the methods discussed in part in [53]). This representation is then stored in short-term (or long-term) memory as appropriate, providing the knowledge of object types and locations to other components which may need it (as described in [55]).

Auditory perception is performed in the form of speech recognition, utilizing CMU Sphinx4 [56,57] to convert spoken words to text before parsing. For the text to be useful to the agent, it is provided to a parsing system which resolves it to actionable behavior or beliefs used by an action management system, also allowing for dialogues about the action capabilities (as with [39]), analogous to the process discussed in [58].

3.3. Reasoning and Goal Management

We assume actions have preconditions and effects, which enable a classical planning problem: given a goal state, and knowledge of how actions impact the world, a series of actions can be selected to achieve the goal. While plans are often selected by their length, we instead perform plan selection based upon a probability of success assessment: when an action or set of actions has been selected, it can be performed by various task-specific or robot-specific functions. As [6] introduces, and we have expanded, action outcomes are monitored and tracked, and so the likelihood of an autonomously generated plan can be computed. Similarly, as we have introduced and will outline, unexpected events imply the presence of a not-yet-measured state, allowing a context to be changed if necessary.

The action execution subsystem maintains a set of executable robot behaviors which it can perform. Upon receiving a goal in the form of a first-order logical predicate (e.g., `at(screw)` or `pickup(caddy)`), the action subsystem is responsible for determining how it can be resolved. Because the goal is in the form of a first-order logical predicate, a wide variety of strategies can be employed to find a series of actions to be taken to accomplish the task. One common approach is use of the planning domain definition language (PDDL) [59], although in our specific implementation we make use of the system outlined in [53]. We modify the existing action selection strategy to select plans based not on their length, but on their cost. We define “cost” here as “success likelihood”, using the strategy to be outlined next. In doing so, the robot selects and performs a plan based on how likely it is to succeed.

3.4. Self-Assessments and Performance Analysis

The robot tracks information about all action outcomes using the technique described in [6]. In this approach, it is assumed that the architecture has the ability to observe the success or failure of an action. In some cases these failures can be directly observed by the architecture, including software failing to execute, or a process taking too long to execute and being canceled. In many cases, however, success must be defined through some observation of the world after a behavior is performed. For example, the `pickup(caddy)` action is known to lead to a state where the “caddy” object will no longer be visible on the table. If it remains on the table, we can safely conclude that the `pickup(caddy)` has somehow failed. This performance can be recorded over a large number of robot operations to provide a statistical assessment of each robot behavior’s likelihood to succeed. Further, as [6] discusses, this takes place using only typical first-order logic planning operations (such as those used by the previously discussed PDDL), and so does not introduce an additional architectural requirement.

We, therefore, expand [6] by using likelihoods of action performance for cost-optimized task planning. We can calculate the success probability of a plan $\pi = \langle a_1, a_2, \dots, a_n \rangle$ as the product of the probability of each action in the plan succeeding: $P(\pi) = \prod_{i=1}^n \text{success}(a_i)$. Note that we assume independence of each plan step as an approximation to keep the problem tractable (it will at times not be accurate as prior steps in a plan may have been taken to improve the likelihood of later steps, for example).

We modified the success/failure book-keeping from [6]. While they focus on updating action success probabilities based on a single action success, we focus on a set of the most recent prior actions in the plan. We accomplish this by viewing the success of action a_i not just as a probability conditioned on its preconditions but also on the success of a “window” W of up to k action predecessors, which allows us to view the preceding sequence of actions as *context C* for the success or failure of a_i . For example, in a plan to grab an object, we monitor and consider the success of each action in the plan (driving to the object, moving the arm, etc.).

If we then encounter a situation after all actions in $W = \langle a_1, a_2, \dots, a_{i-1} \rangle$ are completed successfully but where a_i failed or had an unexpected outcome without there being a context C that included this failure, then we can add a new context $C' = \langle a_1, a_2, \dots, a_i \rangle$ that consists of all the actions in W and the failed action a_i . Note that because a_i failed, the

agent knows that something in the world must be different, but it might not be able to determine the cause (e.g., because it might not be observable for the agent).

For example, the `closeGripper` action is typically highly reliable, unless the gripper has become jammed. Thus, if the `closeGripper` action fails, it may be statistically reasonable to infer that a gripper jam is occurring. This inference is a fairly straightforward task of comparing the events from the current window to the distributions held within the contexts, and selecting the current context as the one which is the most probable. This selection provides knowledge of other action outcome distributions, which may not have been directly observed in the current window.

This knowledge is useful for future action selection—actions which were previously highly reliable can be substituted for otherwise non-optimal behaviors. It is additionally useful for dialogues—if the contexts have been previously recorded and labeled, they can then be used to generate explanations or to describe solutions.

Figure 3 shows a simplified example with two different contexts in C : C_0 , which we have labeled as the fully operational context, and C_1 , which we have labeled as being produced by a jammed gripper. As the agent takes the actions `goToPose("prepare")`, `approach("screw")`, and `gripperGoTo("screw")` (`goToPose(X)` moves the arm to some pre-saved pose X ; `approach(X)` drives the robot up to some landmark associated with X ; `gripperGoTo(X)` identifies X in the scene and moves the gripper towards it. We also use the `closeGripper()` action, which closes the gripper; `pickUp(X)` identifies and attempts to grasp X ; `scoop(X)` attempts to hook and lift X ; and others), each outcome is observed, making C_0 and C_1 indistinguishable in the current window W . However, when the agent attempts `closeGripper()`, a significant difference is observed, making C_1 the more likely context, which has a high probability for `pickUp(caddy)` failing. As a result, the robot selects the otherwise less-performant `scoop("caddy")` action instead to proceed.

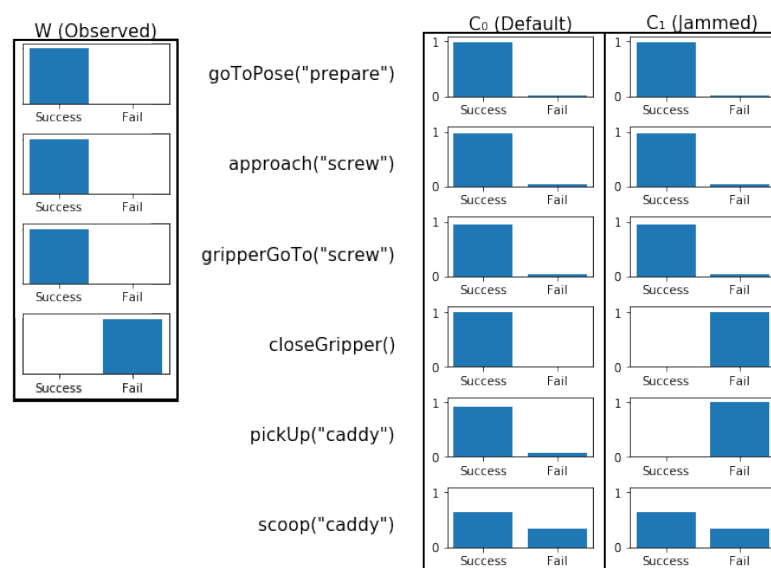


Figure 3. Action outcomes (W) and prior learned contexts C_0 , C_1 .

3.5. A Problem-Solving Example

With this fully integrated system, we can demonstrate the complex interplay between knowledge and reasoning, planning, acting, observing, and dialogue that gives rise to the following dialogue:

H: What is the probability that you can fetch the gearbox top?

R: The probability is 0.

H: Pick up the caddy.

R: I don't think I can pick up, but I can scoop the caddy. [*The robot scoops the caddy.*]

Assuming the agent knows about the past successes of `goToPose`, `approach`, and `gripperGoTo`, as well as the failure of `closeGripper`, the NLU (natural language understanding) pipeline receives and interprets the human query about the probability of the `fetch` action which is known to the agent to be composed of the `approach`, `gripperGoTo`, and `closeGripper` actions, among others. While the probability of many of these actions is high, the probability of `closeGripper` is zero; the requested action is, therefore, guaranteed to fail, which the agent communicates.

The next human instruction triggers a similar set of processes: the request is resolved to the “`pickUp(caddy)`” goal. While performing the action selection, however, the same performance assessment process reveals that the execution of this action is highly unlikely to succeed, which results in a search for an alternative (in general through planning), and the robot finds an action with the effect of holding the caddy, namely `scoop`. While this action would generally have a low chance of success and not be worth attempting, in this case, it is the better of the two options and the initial goal of `pickUp` is replaced with a goal to `scoop` the caddy together with an explanation goal to justify the new behavior to the human.

The full sequence of this demonstration is visible in Figure 4.

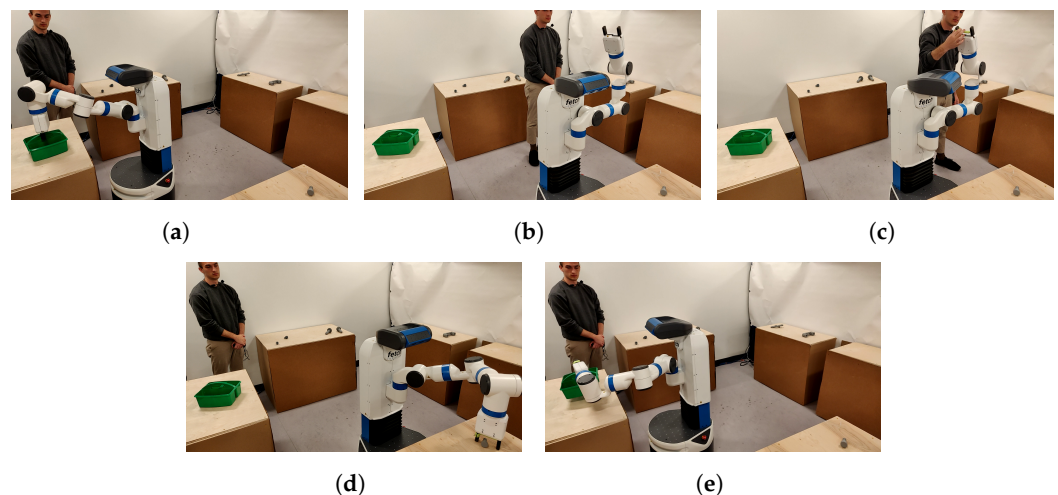


Figure 4. The Fetch prefers the more reliable grasping strategy to grab the caddy (a), and the actions while approaching a new task are in line with expectation (b). The gripper jam is introduced without the knowledge of the agent (c), and so the instructed grasping task fails (d). This experience is used to determine that the “scooping” action should be substituted when grasping the caddy, which it explains and does (e).

4. Case Studies

We next provide several case studies to demonstrate the capabilities resulting from the novel integration of performance assessment with fault detection, reasoning, and recovery planning. The robot and environment are the same as previously described, and the robot knows about several contexts: a fully operational case, a power failure, a large gripper jam (where no objects can be grabbed), and a small gripper jam (where some objects can be grabbed). The length of W has been made unusually short to make it very sensitive ($W = 3$). This value has been chosen for brevity of interaction and demonstration.

4.1. Communicating the Current Context

In our first demonstration case, we introduce an unmitigatable fault in the form of a motor power failure. Although the agent is unable to solve this problem, it does remain capable of providing a dialogue to assist with a human partner’s understanding of how this impact may affect future performance. The parsing and dialogue components enable interactions between a human partner (“H”) and the Fetch robot (“R”), demonstrating that the robot is capable of independent action and self-assessment:

H: Go to pose 'Prepare'. [*The robot arm moves over its head.*]
 H: Approach the caddy. [*The robot drives towards the caddy.*]
 H: What is the probability that you can fetch the gearbox top?
 R: The probability is 0.96.

The motor power is then disconnected from the robot without its knowledge (by pressing the mechanical emergency stop ("e-stop"), motor power is disconnected, but the sensing and computation systems remain online. This fault manifests itself exactly as physically cutting a critical power wire would, without damaging the robot). The instructions continue, but cannot be executed:

H: What is the probability that you can approach the screw?
 R: The probability is 1.0.
 H: Go to pose 'Prepare'. [*The robot does not move.*]
 H: Grab the caddy. [*The robot does not move.*]

The robot cannot determine the specific nature of the failure—although it can observe action failures, it lacks direct knowledge of the power failure. With additional failures, its observations deviate enough from expectations that it is now able to infer we are in a context in which the motor movements consistently fail. When a not-yet-performed action is queried, this can become useful information for the human partner:

H: What is the probability that you can fetch the gearbox top?
 R: The probability is 0.0.

Despite the failure being impossible for the robot to resolve, it has successfully identified its impact, allowing the dialogue component to communicate them to a human partner. Note also that the observed and queried actions are unrelated—the agent is not updating the existing context with this new information, it is shifting to a separate set of experience entirely. This shift provides an inference that other actions have been impacted, despite not being directly observed.

4.2. Optimizing for Success

In the FetchIt! task, the large and small gears are perceptually similar, but because of their different size, the large gear is harder to pick up. This information can be queried:

H: What is the probability that you can fetch the large gear?
 R: The probability is 0.77.
 H: What is the probability that you can fetch the small gear?
 R: The probability is 0.83.

The robot knows that `fetch(largeGear)` and `fetch(smallGear)` both satisfy `pick(gear)` when it receives a new instruction:

H: Pick a gear.

Despite the ambiguity, there is a clear best option. Based on performance assessment on both, the robot selects the most appropriate action:

R: OK. [*The robot grabs the small gear.*]

4.3. Identifying Implications of Failure

The first failure scenario begins with a series of instructions to the robot, with the robot providing its reply:

H: What is the probability that you can approach the small gear?
 R: The probability is 1.0.
 H: What is the probability that you can grab the gearbox top?
 R: The probability is 0.99.
 H: Approach the caddy. [*The robot approaches the caddy.*]
 H: Fetch the screw.

When the first set of interactions concludes, the agent begins a series of behaviors to complete the human-specified goal of fetching a screw for the caddy. At each behavior step, the previously outlined process of action followed by self-assessment takes place. Each step of fetching the screw for the caddy meets with the baseline expectation: `approach(screw)`, `searchFor(screw)`, and `goTo(screw)` are all in-line with expectation. Upon attempting the grasp with the `closeGripper()` action, however, the custom plastic component obstructs the full closing of the grippers.

The agent is now presented the problem of identifying and recovering from failure. We utilized the failure explanation mechanism in [58] in the dialogue component to generate an explanation:

R: I cannot fetch the screw because grasping does not grasp.

In the baseline scenario, closing the gripper is a behavior which is highly likely to succeed. In the current window W , then, it is suddenly improbable that the current context is one of the robot operating correctly. In contrast, among the available contexts, a context where the gripper is jammed is very likely. As a result, the agent now chooses to switch from the original context to this one in which the gripper is jammed, providing it with an alternative pool of experience to sample from. This new context can then be queried to provide updated information to the human partner, despite not having experienced the queried event in the current execution:

H: What is the probability that you can approach the small gear?

R: The probability is 1.0.

H: What is the probability that you can grab the gearbox top?

R: The probability is 0.0.

4.4. Context-Dependent Action Outcomes

The prior gripper jam demonstration could feasibly be addressed using a mechanism which fully disables the grasp action, thus forcing the scoop action to be chosen for the same outcome. With this final demonstration, however, we show that this is not the ideal behavior: there exists a set of cases where the probable outcomes are disrupted within an action such that it is heavily impacted for some cases but minimally impacted for others. In such cases, an action may conditionally remain highly viable.

To demonstrate this case, we construct a new gripper jam scenario: while the gripper still cannot fully close, it can *almost* close and so can still grab some larger objects. Depending on the object, then, grasp may remain the most suitable choice. We have modified this action to conclude with a visual search, allowing grasping failure detection by observing if the object remains on the table after an attempted grasp (though importantly, this observation provides only success/failure information, not failure reasoning information). A series of instructions allow the agent to reason that it is unable to grasp a specific set of objects:

H: Grab the gearbox top. [*The robot grabs the gearbox top.*]

H: Grab the caddy. [*The robot fails to grab the caddy.*]

R: I cannot grab the caddy because 'grasping' does not grasp.

These interactions inform the agent of a unique jamming case: while failure to grasp the caddy is unusual enough to demonstrate some form of failure, the observation that grasping the gearbox top succeeds provides the knowledge that it is unique from the previously demonstrated jamming scenario (recall that all alternative contexts have been available to the agent throughout all demonstrations, and so the prior jamming scenario could have been selected if not for this evidence). From here, we return to the "pick a gear" scenario:

H: Pick a gear. [*The robot grabs the larger of the two gears.*]

The Fetch appropriately chooses the larger of the two gears. Despite being less likely to succeed in typical operation, the context switch provides the knowledge that the larger gear is more likely to succeed. Because the goal of "holding a gear" can be completed with two equally valid plans (grabbing the small or large gear), our performance-optimizing

approach enables the agent to select the strategy which is most likely to succeed. Had a more “naive” approach of disabling the grasp action been used instead, the agent would have had no alternative but to fail to complete the task. Thus, the agent has made the most of a limited set of object interactions to maximize its ability despite the ongoing failure.

5. User Study

As the previous case studies demonstrate, our proposed system integration enables several important capabilities that a robot apprentice working with a human supervisor on a task would need to have to be an effective collaborator. However, it remains to be proven that our approach is actually *preferred*. It is possible that users interacting with a system like ours may strongly not prefer its ability to subvert instruction, even if it does produce a higher chance at success. Previous iterations of the architecture (discussed in more detail in Section 2) have been limited to only taking action corrections as specifically instructed by a human partner (e.g., [60]). As a result it is important to confirm that people interacting with a robot do actually value the ability of a robot to behave as a partner and not simply a tool.

To determine the effectiveness of our approach on human–robot collaboration, we use *functional trust* as an approximate metric, as the impact of trust on human–robot teaming has been well established in the previous literature [61–63], allowing us to use widely used measures of functional trust in an agent to imply perception of an agent as an effective partner for future tasks. While a full-scale real-world user study is beyond the scope of this work, we make use of an online user preference study (based on behavior recorded from the actual deployed system) to demonstrate our system provides functionality which is preferred by potential interactants.

5.1. Methods

Participants. A total of 20 participants who were over 18 and spoke English participated in this study online using the Prolific system. Participants were between 19 and 40 years old ($M = 29.55$, $SD = 6.89$ years). The gender distribution for the sample was: male 50%, female 50%. The ethnic distribution for the sample was: White 75%, Black or African American 20%, Asian 15%, Hispanic 5%. Compensation was USD 2.50.

Conditions. We had five within-subjects conditions showing different ways that the robot responded to failure. In each condition, the participants saw the Fetch performing the “pick up a caddy” task with the “jammed gripper” failure. Each interaction began with the robot receiving an instruction from a human partner to pick up the caddy. The conditions then differed across conditions A through E (transcripts pick up after the “pick up the caddy” command).

Robot A. This was the baseline failure condition. The robot failed with no acknowledgment of the failure. This robot did not use our mechanism. The video transcript was:

R: OK. [*The robot fails to grab the caddy.*]

Robot B. This robot failed but acknowledged the failure afterwards. It does not use our mechanism. The video transcript was:

R: OK. [*The robot fails to grab the caddy.*]

R: I cannot pick up the caddy.

Robot C. This robot failed, acknowledged the failure, and explained why it failed. This robot used a modified version of our mechanism. The context-switching mechanism explicitly associates each context with a symbolic representation of a state. These are in the same form that the explanation mechanism introduced in [58] makes use of. We can, therefore, use the combination of these two mechanisms to provide this more verbose failure explanation. The video transcript was:

R: OK. [*The robot fails to grab the caddy.*]

R: I cannot pick up the caddy because my gripper is jammed.

Robot D. In this case, the robot does not fail. It knew that it would not be able to complete the pick up action, and told this to the person instead of attempting to execute it. The person then gave it a new command, which it executed properly. This robot utilized our mechanism. The video transcript was:

R: I don't think I can pick up.

H: Scoop the caddy.

R: OK. [*The robot scoops the caddy.*]

Robot E. Here again, the robot does not fail. It identified that it could not pick up the caddy and proactively performed a replacement action. This robot utilized our mechanism. The video transcript was:

R: I don't think I can pick up, but I can scoop the caddy. [*The robot scoops the caddy.*]

The comparison of Robots A, B, and C evaluates our explanation mechanism vs. naive explanation mechanisms. The comparison of Robots A, D, and E evaluates action substitution vs. failure. We predict that the conditions which utilize our mechanism (Robots C, D, and E), will be trusted more and be ranked higher than the conditions which do not utilize our mechanisms (i.e., Robots A and B).

Materials. The study was undertaken using Qualtrics and distributed through Prolific. The videos featured a Fetch robot performing the “pick up a caddy” task with the “jammed gripper” failure. Robots C, D, and E utilized the mechanisms described in this paper.

Procedure. After providing informed consent, participants read an introductory text explaining that they were about to watch a video of the robot completing the task for “pick up the caddy”, followed by a researcher coming in, adding a bracket to the robot’s gripper that would prevent it from being able to close the gripper, and then asking the robot to pick up the caddy again. The robot fails the second time because it cannot fully close its gripper. Participants watched this introduction video, and were then told that they would watch and evaluate five more instances of the robot attempting to pick up the caddy.

Participants always saw the baseline failure, Robot A, as the first experimental condition. They then answered an attention check question, the six trust questions, and an open-ended question about what factors affected their answers. They then saw the remaining robot failure conditions, Robots B, C, D, and E, in a random order and answered the same set of questions for every failure video.

Once the participants had seen all the failure conditions, they were told that they would be ranking the different robots from each failure condition directly against each other. They were asked to rank the robots in terms of the ease of operation, proactivity, interaction level, and understanding. Finally, the participants answered an overall attention check question, a short demographics questionnaire, and then received their compensation. All procedures were approved by our institution’s IRB.

Measures. To measure trust, we used the Reliance Intention Scale, a six-question measure of trust scale developed by [64]. The questions were answered on a 7-point “Strongly disagree” to “Strongly agree” Likert scale, and the six questions were averaged across to give one overall trust score per robot per person. To determine the robot rankings, the participants chose which of the five robots they wanted to put as rank #1, rank #2, etc, with the #1 spot being considered the best for that question.

5.2. Results

Quantitative Results. A one-way repeated measures ANOVA found a significant effect of Robot on the trust scores, $F(4) = 16.13, p < 0.0001$. A series of pairwise T-tests with Bonferroni corrections found the following significant comparisons: Robots E ($M = 4.53, SD = 1.61$), D ($M = 4.01, SD = 1.39$), and C ($M = 3.82, SD = 1.41$) were all trusted more than Robot A ($M = 2.33, SD = 1.32$) and Robot B ($M = 2.73, SD = 1.17$) (see Figure 5a). There was no significant difference between Robots E, D, or C and each other, and no significant difference between Robots B and A and each other.

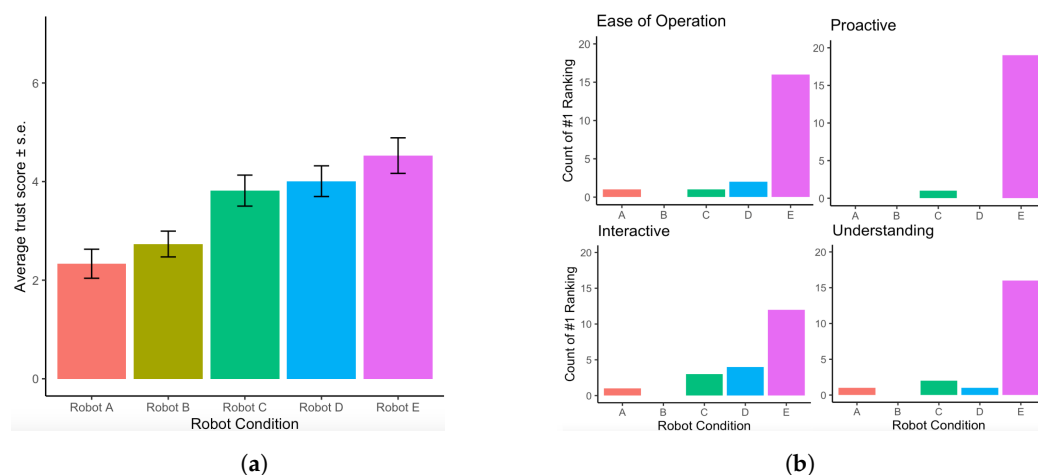


Figure 5. Results of quantitative analysis. (a) Histogram of average trust scores for each robot condition with standard error error bars. We observe that Robot E scores highest in trust across all conditions. (b) Counts of which robot was ranked in the #1 spot for each of the four ranking questions. Robot E (far right) is consistently ranked #1 by a large margin.

For the ranking questions, Robot E was consistently ranked as the #1 robot for all of the questions (presented in Figure 5b). For all the data, see Appendix A. Robot D was consistently ranked #2 for all four questions, Robot C was ranked #3, Robot B was ranked #4, and Robot A, our baseline failure condition, was consistently ranked #5 for all four ranking questions.

User Responses. In their qualitative answers, the instantiations of our mechanism were mentioned specifically when participants were describing their positive views of Robots E, D, and C. Here, we provide examples of the feedback our robots received for each of the questions participants were asked. When describing the reasoning behind their trust ratings for Robot E, one participant said “It problem-solved very well and found a way to pick up the caddy despite the issues”. Another said “The fact that the robot can assess what it cannot do while also providing an alternative solution to the problem makes it the most trustworthy of all that I have seen today. It is smart and adaptable”. Our mechanism’s use in Robot E was also cited as being the reason most people ranked it at the #1 spot for our four rankings. One participant’s explanation for what factors affected their “ease of operation” rank was “How aware the robot was or how adaptive it could be. The Robot E actually proposed its own solution, which is very impressive”. For proactivity, one person said “E took the initiative to find its own solution”. For interaction, both Robots E and D were discussed as in “I found that the robots who could explain why they could not complete the task, or who could understand alternative motives were the most interactive”. Finally, for understanding, “Robot E was the most communicative and able to express what it was doing and how it was problem solving”.

Our mechanism was also recognized in Robots D and C, and contributed to the positive ratings those robots received as well. For Robot D, one participant described how their trust ratings were determined based on the fact that “Though the robot was able to successfully follow further instruction, it was not able to suggest the new idea on its own”. For Robot C, one participant said that “The robot’s inability to perform the simple task of picking up the caddy made me view it negatively, but when it gave a specific reason for why it couldn’t pick up the caddy (the gripper being jammed), this made me view it a little more positively, because it made me feel that it might be easier to troubleshoot”.

These reflections of how our mechanism positively influenced the trust and robot rankings are opposed to the negative responses garnered by Robots A and B, which did not use our mechanism. For example, Robot A was rated with low trust because “The robot’s failure and lack of [recognition] of the failure affected my decisions”. For Robot B, which did acknowledge that it failed but not why, “This robot did not ask if there was a better

way for it to complete the task, it just stated that it could not complete it”, and “It could respond to the fact that it couldn’t do the task, but it didn’t state why it couldn’t which makes it less reliable”. Our quantitative results show that robots using our mechanism were rated significantly more favorably than robots that did not. Our qualitative results show that the participants recognize that it was specifically because of our mechanism that this difference was achieved.

6. Discussion and Future Work

In our study, we showed that the proposed integrated system, in various forms, resulted in participants trusting a robot more than when the robot was not controlled by our proposed system. Because trust is an important component to many aspects of human–robot interaction, we consider this a success. Additionally, we found that Robot E, the robot that utilized our system to make proactive action decisions and to avoid failures, was considered the best robot in terms of ease of operation, proactivity, interaction, and understanding levels. Robots D and C, the robots that used our system to a lesser degree, were ranked second and third, respectively. This highlights how our proposed mechanisms and integration results in increased robot preference among potential users.

It is important to note that we have not introduced additional constraints to the existing robot planning/action execution domain. We leverage components already present in the robot planning domain, and make use of measurements which may not explicitly be in the traditional domain but are otherwise easy to obtain (action success/failure). For this reason, our work remains as general as the work it builds upon, and can, therefore, be applied to any other agent planning/execution domain without loss of the generalizability of these approaches.

The agent, depending upon experience to inform new decisions, presents a need to collect data prior to use. Our approach can benefit here from future advances in reinforcement learning, which is faced with the same data-collection problem. We can also benefit from the manner in which we model action outcomes. The knowledge that some action a on object o has an outcome is distinct from the outcome of a on o' , and so these are represented as two unique datapoints. However, the knowledge of the similarity between o and o' may provide a starting point for learning the unique action outcomes of the two objects.

7. Conclusions

We have presented a novel integrated system for a “robot apprentice” that utilized performance assessment methods paired with introspection mechanisms to enable the statistics-based self-assessment of action outcomes used for identifying task execution failures, communicating them to humans, and if possible, autonomously resolving them. We presented task-based dialogue interactions with a human supervisor using several case studies that showed the flexibility and autonomy of the system to cope with unexpected events as one would expect of an apprentice. Furthermore, an online user study showed a strong human preference for the capabilities we enabled in our integrated system compared to variations of the system that lacked some of the features. We believe that the proposed system is an important next step for developing versatile collaborative robot systems that can enable collaborative interactions that go beyond humans micromanaging the robot.

Author Contributions: C.T.: conceptualization, implementation, writing, and a portion of the user study; T.L.: conducted the user study and provided writing of the analysis; T.F.: conceptualization and some implementation; M.S.: conceptualization, writing, and advising. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Office of Naval Research grants #N00014-18-1-2831 and #N00014-18-1-2503.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki, and approved by the Institutional Review Board (or Ethics Committee) of Tufts University (SBER).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Valsiner, J.; Van der Veer, R. *The Social Mind: Construction of the Idea*; Cambridge University Press: Cambridge, UK, 2000.
2. Pezzulo, G.; Dindo, H. What should I do next? Using shared representations to solve interaction problems. *Exp. Brain Res.* **2011**, *211*, 613–630. [[CrossRef](#)] [[PubMed](#)]
3. Cantrell, R.; Schermerhorn, P.; Scheutz, M. Learning actions from human-robot dialogues. In Proceedings of the 2011 RO-MAN, Atlanta, GA, USA, 31 July–3 August 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 125–130.
4. Scheutz, M.; Schermerhorn, P.; Kramer, J.; Anderson, D.C. First steps toward natural human-like hri. *Auton. Robot.* **2006**, *22*, 411–423. [[CrossRef](#)]
5. Huffman, S.B.; Laird, J.E. Flexibly instructable agents. *J. Artif. Intell. Res.* **1995**, *3*, 271–324. [[CrossRef](#)]
6. Frasca, T.M.; Scheutz, M. A framework for robot self-assessment of expected task performance. *IEEE Robot. Autom. Lett.* **2022**, *7*, 12523–12530. [[CrossRef](#)]
7. Majji, M.; Rai, R. Autonomous task assignment of multiple operators for human robot interaction. In Proceedings of the 2013 American Control Conference, Washington, DC, USA, 17–19 June 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 6454–6459.
8. Xu, Y.; Dai, T.; Sycara, K.; Lewis, M. Service level differentiation in multi-robots control. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 2224–2230.
9. Srivastava, V.; Surana, A.; Bullo, F. Adaptive attention allocation in human-robot systems. In Proceedings of the 2012 American Control Conference (ACC), Montreal, QC, Canada, 27–29 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 2767–2774.
10. Chien, S.Y.; Lewis, M.; Mehrotra, S.; Brooks, N.; Sycara, K. Scheduling operator attention for multi-robot control. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 473–479.
11. Shi, H.; Xu, L.; Zhang, L.; Pan, W.; Xu, G. Research on self-adaptive decision-making mechanism for competition strategies in robot soccer. *Front. Comput. Sci.* **2015**, *9*, 485–494. [[CrossRef](#)]
12. Dai, T.; Sycara, K.; Lewis, M. A game theoretic queueing approach to self-assessment in human-robot interaction systems. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 58–63.
13. Lewis, M. Human interaction with multiple remote robots. *Rev. Hum. Factors Ergon.* **2013**, *9*, 131–174. [[CrossRef](#)]
14. Ardón, P.; Pairet, E.; Petillot, Y.; Petrick, R.P.; Ramamoorthy, S.; Lohan, K.S. Self-assessment of grasp affordance transfer. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; IEEE: Piscataway, NJ, USA, 2020; pp. 9385–9392.
15. Ardón, P.; Cabrera, M.E.; Pairet, E.; Petrick, R.P.; Ramamoorthy, S.; Lohan, K.S.; Cakmak, M. Affordance-aware handovers with human arm mobility constraints. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3136–3143. [[CrossRef](#)]
16. Chen, F.; Sekiyama, K.; Huang, J.; Sun, B.; Sasaki, H.; Fukuda, T. An assembly strategy scheduling method for human and robot coordinated cell manufacturing. *Int. J. Intell. Comput. Cybern.* **2011**, *4*, 487–510.
17. Conlon, N.; Szafir, D.; Ahmed, N.R. Investigating the Effects of Robot Proficiency Self-Assessment on Trust and Performance. *arXiv* **2022**, arXiv:2203.10407.
18. Lefort, M.; Gepperth, A. Active learning of local predictable representations with artificial curiosity. In Proceedings of the 2015 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob), Providence, RI, USA, 13–16 August 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 228–233.
19. Shi, H.; Lin, Z.; Zhang, S.; Li, X.; Hwang, K.S. An adaptive decision-making method with fuzzy Bayesian reinforcement learning for robot soccer. *Inf. Sci.* **2018**, *436*, 268–281.
20. Jauffret, A.; Grand, C.; Cuperlier, N.; Gaussier, P.; Tarroux, P. How can a robot evaluate its own behavior? A neural model for self-assessment. In Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 4–9 August 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 1–8.
21. Alami, R.; Chatila, R.; Clodic, A.; Fleury, S.; Herrb, M.; Montreuil, V.; Sisbot, E.A. Towards human-aware cognitive robots. In Proceedings of the Fifth International Cognitive Robotics Workshop (the AAAI-06 Workshop on Cognitive Robotics), Boston, MA, USA, 16–17 July 2006.
22. Alami, R.; Clodic, A.; Montreuil, V.; Sisbot, E.A.; Chatila, R. Toward Human-Aware Robot Task Planning. In Proceedings of the AAAI Spring Symposium: To Boldly Go Where No Human-Robot Team Has Gone Before, Stanford, CA, USA, 27–29 March 2006; pp. 39–46.
23. Kwon, W.Y.; Suh, I.H. Planning of proactive behaviors for human–robot cooperative tasks under uncertainty. *Knowl.-Based Syst.* **2014**, *72*, 81–95.
24. Koppula, H.S.; Jain, A.; Saxena, A. Anticipatory planning for human-robot teams. In *Experimental Robotics*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 453–470.

25. Buisan, G.; Sarthou, G.; Alami, R. Human aware task planning using verbal communication feasibility and costs. In Proceedings of the International Conference on Social Robotics, Golden, CO, USA, 14–18 November 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 554–565.
26. Liu, C.; Hamrick, J.B.; Fisac, J.F.; Dragan, A.D.; Hedrick, J.K.; Sastry, S.S.; Griffiths, T.L. Goal Inference Improves Objective and Perceived Performance in Human-Robot Collaboration. *arXiv* **2018**, arXiv:1802.0178.
27. Devin, S.; Clodic, A.; Alami, R. About decisions during human-robot shared plan achievement: Who should act and how? In Proceedings of the International Conference on Social Robotics, Tsukuba, Japan, 22–24 November 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 453–463.
28. Hoffman, G.; Breazeal, C. Cost-based anticipatory action selection for human–robot fluency. *IEEE Trans. Robot.* **2007**, *23*, 952–961.
29. Hoffman, G.; Breazeal, C. Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team. In Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction, Arlington, VA, USA, 8–11 March 2007; pp. 1–8.
30. Kim, Y.; Yoon, W.C. Generating task-oriented interactions of service robots. *IEEE Trans. Syst. Man, Cybern. Syst.* **2014**, *44*, 981–994. [\[CrossRef\]](#)
31. Wolfe, J.A.; Marthi, B.; Russell, S. Combined Task and Motion Planning for Mobile Manipulation. In Proceedings of the 20th International Conference on Automated Planning and Scheduling, ICAPS 2010, Toronto, ON, Canada, 12–16 May 2010.
32. Foehn, P.; Romero, A.; Scaramuzza, D. Time-optimal planning for quadrotor waypoint flight. *Sci. Robot.* **2021**, *6*, 1221. [\[CrossRef\]](#)
33. Canal, G.; Alenyà, G.; Torras, C. Adapting robot task planning to user preferences: An assistive shoe dressing example. *Auton. Robot.* **2019**, *43*, 1343–1356. [\[CrossRef\]](#)
34. Kulkarni, A.; Zha, Y.; Chakraborti, T.; Vadlamudi, S.G.; Zhang, Y.; Kambhampati, S. Explicable Planning as Minimizing Distance from Expected Behavior. In Proceedings of the AAMAS Conference Proceedings, Montreal, QC, Canada, 13–17 May 2019; pp. 2075–2077.
35. Forer, S.; Banisetty, S.B.; Yliniemi, L.; Nicolescu, M.; Feil-Seifer, D. Socially-aware navigation using non-linear multi-objective optimization. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–9.
36. Nissim, R.; Brafman, R.I. Cost-Optimal Planning by Self-Interested Agents. In Proceedings of the Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, Bellevue, WA, USA, 14–18 July 2013; pp. 732–738. [\[CrossRef\]](#)
37. Lee, C.Y.; Shen, Y.X. Optimal planning of ground grid based on particle swarm algorithm. *Int. J. Eng. Sci. Technol. (IJEST)* **2009**, *3*, 30–37.
38. Khandelwal, P.; Yang, F.; Leonetti, M.; Lifschitz, V.; Stone, P. Planning in action language BC while learning action costs for mobile robots. In Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling, Portsmouth, NH, USA, 21–26 June 2014.
39. Frasca, T.M.; Krause, E.A.; Thielstrom, R.; Scheutz, M. “Can you do this?” Self-Assessment Dialogues with Autonomous Robots Before, During, and After a Mission. *arXiv* **2020**, arXiv:2005.01544.
40. Norton, A.; Admoni, H.; Crandall, J.; Fitzgerald, T.; Gautam, A.; Goodrich, M.; Saretsky, A.; Scheutz, M.; Simmons, R.; Steinfeld, A.; et al. Metrics for robot proficiency self-assessment and communication of proficiency in human-robot teams. *ACM Trans. Hum.-Robot Interact. (THRI)* **2022**, *11*, 20.
41. Baraglia, J.; Cakmak, M.; Nagai, Y.; Rao, R.P.; Asada, M. Efficient human-robot collaboration: When should a robot take initiative? *Int. J. Robot. Res.* **2017**, *36*, 563–579.
42. Nikolaidis, S.; Kwon, M.; Forlizzi, J.; Srinivasa, S. Planning with verbal communication for human-robot collaboration. *ACM Trans. Hum.-Robot Interact. (THRI)* **2018**, *7*, 22. [\[CrossRef\]](#)
43. St. Clair, A.; Mataric, M. How robot verbal feedback can improve team performance in human-robot task collaborations. In Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, Portland, OR, USA, 2–5 March 2015; pp. 213–220.
44. Unhelkar, V.V.; Li, S.; Shah, J.A. Decision-making for bidirectional communication in sequential human-robot collaborative tasks. In Proceedings of the 2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Cambridge, UK, 23–26 March 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 329–341.
45. Beer, J.M.; Fisk, A.D.; Rogers, W.A. Toward a framework for levels of robot autonomy in human-robot interaction. *J. Hum.-Robot Interact.* **2014**, *3*, 74. [\[CrossRef\]](#) [\[PubMed\]](#)
46. Fong, T.; Thorpe, C.; Baur, C. Collaboration, dialogue, human-robot interaction. In *Robotics Research*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 255–266.
47. Langley, P. Explainable agency in human-robot interaction. In Proceedings of the AAAI Fall Symposium Series, Arlington, VA, USA, 17–19 November 2016.
48. Duffy, B. Robots social embodiment in autonomous mobile robotics. *Int. J. Adv. Robot. Syst.* **2004**, *1*, 17. [\[CrossRef\]](#)
49. Wagner, A.R. Robots that stereotype: Creating and using categories of people for human-robot interaction. *J. Hum.-Robot Interact.* **2015**, *4*, 97. [\[CrossRef\]](#)
50. Kaupp, T.; Makarenko, A.; Durrant-Whyte, H. Human–robot communication for collaborative decision making—A probabilistic approach. *Robot. Auton. Syst.* **2010**, *58*, 444–456. [\[CrossRef\]](#)

51. Fetch Robotics, Inc. FetchIt! Challenge. Available online: <https://opensource.fetchrobotics.com/competition> (accessed on 1 December 2023).
52. Wise, M.; Ferguson, M.; King, D.; Diehr, E.; Dymesich, D. Fetch and freight: Standard platforms for service robot applications. In *Proceedings of the Workshop on Autonomous Mobile Service Robots*, New York, NY, USA, 11 July 2016.
53. Scheutz, M.; Williams, T.; Krause, E.; Oosterveld, B.; Sarathy, V.; Frasca, T. An overview of the distributed integrated cognition affect and reflection DIARC architecture. In *Cognitive Architectures*; Springer: Cham, Switzerland, 2019; pp. 165–193.
54. Scheutz, M.; Thielstrom, R.; Abrams, M. Transparency through Explanations and Justifications in Human-Robot Task-Based Communications. *Int. J.-Hum.-Comput. Interact.* **2022**, *38*, 1739–1752. [[CrossRef](#)]
55. Briggs, G.; Scheutz, M. Multi-modal belief updates in multi-robot human-robot dialogue interaction. In *Proceedings of the 2012 Symposium on Linguistic and Cognitive Approaches to Dialogue Agents*; University of Birmingham: Birmingham, UK, 2012; Volume 47.
56. Walker, W.; Lamere, P.; Kwok, P.; Raj, B.; Singh, R.; Gouvea, E.; Wolf, P.; Woelfel, J. *Sphinx-4: A Flexible Open Source Framework for Speech Recognition*; Sun Microsystems, Inc.: Santa Clara, CA, USA, 2004.
57. Lamere, P.; Kwok, P.; Gouvea, E.; Raj, B.; Singh, R.; Walker, W.; Warmuth, M.; Wolf, P. The CMU SPHINX-4 speech recognition system. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2003)*, Hong Kong, China, 6–10 April 2003; Volume 1, pp. 2–5.
58. Thielstrom, R.; Roque, A.; Chita-Tegmark, M.; Scheutz, M. Generating Explanations of Action Failures in a Cognitive Robotic Architecture. In *Proceedings of the 2nd Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence*, Dublin, Ireland, 18 December 2020; pp. 67–72.
59. Fox, M.; Long, D. PDDL2. 1: An extension to PDDL for expressing temporal planning domains. *J. Artif. Intell. Res.* **2003**, *20*, 61–124. [[CrossRef](#)]
60. Thierauf, C.; Thielstrom, R.; Oosterveld, B.; Becker, W.; Scheutz, M. “Do This Instead”—Robots That Adequately Respond to Corrected Instructions; Association for Computing Machinery: New York, NY, USA, 22 September 2023. [[CrossRef](#)]
61. Rossi, A.; Dautenhahn, K.; Koay, K.L.; Walters, M.L. The impact of peoples’ personal dispositions and personalities on their trust of robots in an emergency scenario. *Paladyn J. Behav. Robot.* **2018**, *9*, 137–154. [[CrossRef](#)]
62. Lee, J.D.; See, K.A. Trust in automation: Designing for appropriate reliance. *Hum. Factors* **2004**, *46*, 50–80. [[CrossRef](#)]
63. Freedy, A.; DeVisser, E.; Weltman, G.; Coeyman, N. Measurement of trust in human-robot collaboration. In *Proceedings of the 2007 International Symposium on Collaborative Technologies and Systems*, Orlando, FL, USA, 21–25 May 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 106–114.
64. Lyons, J.B.; Guznov, S.Y. Individual differences in human–machine trust: A multi-study look at the perfect automation schema. *Theor. Issues Ergon. Sci.* **2019**, *20*, 440–458. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.