

Model Reconciliation through Explainability and Collaborative Recovery in Assistive Robotics

Britt Besch^{1*,2}, Tai Mai¹, Jeremias Thun¹, Markus Huff^{3,4}, Jörn Vogel¹, Freck Stulp¹, Samuel Bustamante^{1,5}

Abstract—Whenever humans and robots work together, it is essential that unexpected robot behavior can be explained to the user. Especially in applications such as shared control — as the name may imply — the user and the robot must *share* the same model of the objects in the world, and the actions that can be performed on these objects.

We achieve this with a model reconciliation framework. We use a Large Language Model (LLM) to predict and explain differences between the robot’s and human’s mental models, without requiring a formal model of the user. The framework also resolves model divergence by allowing the human to correct the robot after the explanation. We provide an implementation in assistive robotics and conduct experiments with a real wheelchair-based mobile manipulator and its digital twin.

I. INTRODUCTION

Whenever humans and robots work together — such as assistive robots for people with disabilities or cobots in industry — it is essential that unexpected robot behavior can be explained to the user, as conveying beliefs used in decision-making builds trust [1]. For instance, a robot may wrongly believe that an appliance is open even though it is closed, and thus fail to provide the user with the support they need. If the robot can explain its beliefs about the appliance state, the human can understand why the task failed and correct the robot’s knowledge to recover.

To address this challenge, we draw on mental modeling and AI planning by conceptualizing explanations as *model reconciliation*: the robot aims to bring the human to a shared understanding of an error situation by comparing the robot’s and human’s internal models [2]–[4].

We consider model reconciliation in shared control of assistive robots for people with motor impairments performing activities of daily living. We propose a framework that uses an LLM workflow [5]: from the robot’s models and the human’s natural-language queries, it predicts and explains differences between their mental models. The framework explains predicted differences between user and robot knowl-

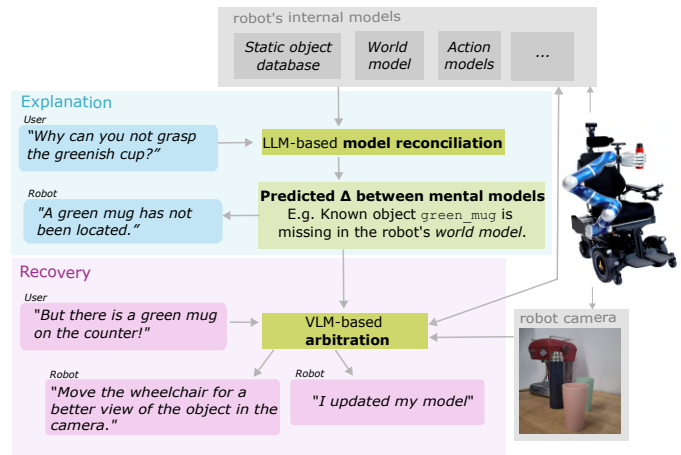


Fig. 1. Overview of the model reconciliation framework including the bidirectional updating process between human and robot, using foundation models.

edge (e.g., world model) without requiring the robot to know the user’s full mental model.

As an example with our wheelchair-based mobile manipulator EDAN [6], [7] in Fig. 1, the human inquires why the robot cannot grasp a *greenish cup*. The workflow identifies that the human is referring to an object known to the robot as *green_mug*. This object is not present in the robot’s model of the world, and thus it provides an explanation in natural language.

Furthermore, our framework addresses model divergence after explanation by allowing the human to correct the robot. We achieve this through an arbitration module with a Vision-Language Model (VLM) that helps the human recover. Continuing the example in Fig. 1, the workflow can give robot-specific instructions, such as moving the wheelchair for a better view, or update the robot’s model by directly adding the *green_mug*.

To summarize, we aim to 1) enable systems to explain their behavior, allowing the human to understand robotic beliefs and 2) create a workflow for recovering from such unexpected situations, particularly when the robot is mistaken, for instance due to perceptual errors. The paper makes contributions on three levels: **firstly**, on a theoretical level by proposing a bi-directional model reconciliation framework for shared control in assistive robotics, encompassing the robot updating the human and the human updating the robot’s model. **Secondly**, on a methodological level via the usage of

¹ German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC), Münchener Str. 20, 82234 Weßling, Germany.
² University of Cambridge, Department of Engineering, Cambridge, UK.
³ Universität Tübingen, Department of Psychology, Tübingen, Germany.
⁴ Leibniz-Institut für Wissensmedien (IWM), Tübingen, Germany.
⁵ Technical University of Munich, Garching, Germany.
 * Work done while at DLR.
 This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy by means of the project SMILE-AI.
 Correspondence: {bb720 (at) cam.ac.uk} & {samuel.bustamante (at) dlr.de}

foundation model workflows that have been augmented with information retrieved from a set of robot’s modules. **Thirdly**, on an experimental level by providing an implementation and a set of experiments in daily living scenarios with our assistive robot, including the real robot and also a digital twin.

II. RELATED WORK

A. Background: mental models and theory of mind

To motivate our framework, we draw insights from the social sciences, inspired by Miller [8]. A good explanation must be relevant not only to the explainee’s questions but also to their mental model. Hence, we require a model of both the explanatory agent and the explainee (we define a mental model as an internal representation of the current state of affairs [9]), and the concept of *Theory of Mind* [8]—the ability to attribute mental states (e.g., beliefs) to oneself and others. This enables agents to predict and interpret others’ behavior from their mental models. It also underlies *first-order false beliefs* [10]: that another can hold an incorrect belief about a fact, which matters when user and robot disagree about the world state (e.g., green mug in Fig. 1).

B. Explanations as Model Reconciliation

Agent explanations based on mental models follow extensive prior research. The AI planning community coined the term “*model reconciliation*” [4] to describe explainability approaches that combine the robot’s model and its model of the human’s model as symbolic representations, and provide their differences as explanations. Several derived works exist, e.g., [3], [11], [12].

However, approaches for model reconciliation often focus on explaining the *optimality* of a plan to the human [12], and make the assumption that the human’s mental model is known to the robot, which is not realistic in assistive robotics as it is cumbersome for the human to provide the entirety of their mental model. Furthermore, many robots with perception systems (e.g. [13]) have dynamic world models that can lead to discrepancies about the world state between the human and the robot models, as in the example in Fig. 1. In some cases, the human may not even know that an object or an action is unknown by the robot.

We take inspiration from previous methods where (1) the robot infers the human model with a dialogue based on a set of rules [11], and (2) explanations reconcile the difference between robot’s and human’s entire knowledge bases, rather than just the optimality of a robot plan [12]. Our approach introduces an LLM workflow that exploits its rich semantic understanding of the user’s query and robot representations to predict the difference between the human and the robot mental models (such as object databases and world models). This workflow provides explanations without the need to (1) explicitly represent the human mental models nor (2) limit the explanations to a plan’s optimality, for instance by allowing to explain divergent initial world states. Our method aligns with a recent workshop paper in which an LLM

identifies facts responsible for model misalignment [14], and further adds an assistive robotics implementation and evaluation.

C. Collaborative Failure Recovery

The methods above explain robot beliefs; when the robot’s model is wrong (misaligned with ground truth), the human can understand but not yet correct it, so failure recovery is not guaranteed.

Collaborative failure recovery methods mainly have the robot detect failures and ask the human for help, typically with the human performing manual tasks (e.g., relocating objects for better detection) [15]–[17]. [18] allows the human to add knowledge via demonstrations for collaborative task decisions and recovery behavior, and [19] allows the human to intervene via speech with limited vocabulary to guide trajectory assistance for a dressing task. More recent work by [20] conducts an uncertainty analysis in robot action plans and uses an LLM to actively query the human for additional information collection to avoid task execution if there is incomplete knowledge.

Yet, approaches that allow for initiation of failure recovery or active intervention by the human are rare. The framework by [21] uses an LLM to integrate human initiated corrections into task re-planning and integrates feedback from VLMs for error diagnosis. [22] further distills the human’s feedback into a reusable knowledge base to enhance performance in novel settings. A recent workshop paper [14] proposes a concept with LLMs, where the human is able to provide a verbal interruption to correct the robot, and the LLM identifies the facts responsible for misalignment through a model reconciliation framework. To our knowledge, prior work does not combine an explanatory pipeline with collaborative failure recovery in a model reconciliation framework.

III. MODEL RECONCILIATION FRAMEWORK

A. Assumptions

Robot control: We consider shared-control systems where human and system jointly manage the robot, in an assistive robotics setup (e.g., [23]), with the robot scheduling one action at a time, guided by the user.

Mental models: We assume that the human has a mental model of the world as well as of the robot’s capabilities. Further, we assume that the robot has three models: (1) a database with (static) general knowledge about objects and actions; (2) a world model, which represents the robot’s beliefs about the current state of the world, such as specific object instances and their location (see [24] for examples); (3) and an action model based on preconditions and effects (e.g. [25]). Section III-D provides details of the implementation we used. Finally, we also assume that the symbols in those models are semantically grounded in the English language. As an example, a robot may represent the missing green object in Fig. 1 with a symbol such as `mug_green$2` or `cup_green_id34`, instead of `object_blob`. In this paper, we use the notation `object$id` for object instances and `action_verb` for actions.

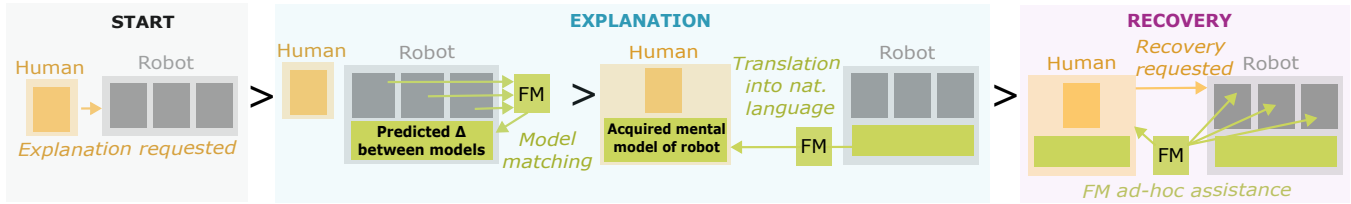


Fig. 2. Evolution of the mental models of human and robot during our model reconciliation framework. Yellow boxes represent the user’s model. Grey boxes represent the robot’s models.

B. Sources of model divergence

When robot and human are manipulating objects in shared control, there are two main dimensions along which the model of the human and the robot may diverge, related to

- knowledge about **objects** themselves (O) vs about **actions** that can be performed on objects (A);
- **general** knowledge about objects/actions (G), e.g. “apples can be picked up” vs. **specific** knowledge about objects and the performable action in the *current* world state (S), e.g. “there is an apple on the table in front of me, but I am currently unable to pick it up”.

From the permutation of these two dimensions, the following four model divergences can be derived:

- **D_{GO}**: The human thinks the robot knows about an object (e.g. a thermos bottle), but it does not (general knowledge about objects).
- **D_{SO}**: The human thinks the robot observes an instance of an object in the current scene (e.g. a specific `lab_thermos$id`), but it does not (specific knowledge about object perception in the current scene).
- **D_{GA}**: The human thinks the robot is able to perform a certain action with an object in general (e.g. pour from bottles), but it is not (general knowledge about actions on objects).
- **D_{SA}**: The human thinks the robot is able to perform a specific action with an object in the current scene (e.g. `execute pour_from` from a `lab_thermos$7` into a `mug$128`), but it is currently not due to a missing symbolic precondition (e.g. if it believes `lab_thermos$37` must be grasped first) (knowledge about preconditions of actions on objects in the current scene).

We further add:

- **FD**: A false divergence, i.e., a human query may imply a divergence, but in reality there is no difference between mental models (e.g. if the human asks why can the robot not pick up `lab_thermos$37`, but the robot can).

C. Model reconciliation

A model reconciliation framework for shared control must thus be able to provide explanations so that the human can identify which of the five divergences (**D_{GO}**, **D_{SO}**, **D_{GA}**, **D_{SA}**, **FD**) between their model and that of the robot causes the misunderstanding. And it must provide reconciliation methods to help resolve the divergence, if possible. We divide

this procedure into three stages (start, explanation, recovery) and illustrate the evolution of the mental models in Fig. 2.

1) *Start*: At the start, both parties are unaware of the other’s mental models, and the human asks the robot for an explanation in natural language. We assume queries of the form “why cannot I/cannot you {action} {with an object}?”; the template does not need to be followed exactly, as language models parse the query. Examples: “why can you not grasp?”; “Hey robot, I cannot open the drawer, what is the issue?”

2) *Explanation*: First, we extract the actions and objects (including adjectives such as “greenish”) from the query using an LLM. Then, we iteratively search in the robot models to *match* the user query actions (and, if provided, objects and adjectives) in natural language. Crucial in this step is that LLMs provide a robust semantic interpretation, both of the query and of the robot model contents.

Once a match is found that explains the disagreement, this is communicated to the human in natural language. We note in Fig. 2 that this explanation updates the mental model that both parties have of each other, as (1) the human updates their own internal model of the robot; and (2) the robot implicitly estimates a human model, as it determines the delta between the models (e.g. the robot learns that the human expects some greenish cup device in Fig. 1).

3) *Recovery procedure*: Provided with an explanation, the human decides if they want to accept it and carry on, or provide a natural language rebuttal to the robot instead. For instance, if the robot explains a drawer cannot be opened because it believes it is already open, the user can rebut with “But the drawer is actually closed!”. In this case, the model is either overwritten directly by the estimated difference (in the simplest case), or the robot provides ad hoc methods to recover (e.g. suggesting robot base movements to improve the perspective of perception methods).

We note that if the divergence concerns general knowledge, such as teaching new skills and objects to the robot (**D_{GO}** & **D_{GA}**), the user may not be able to recover by themselves, as providing this knowledge often requires expert modules. In this paper we focus only on recovery of divergences of specific knowledge, namely **D_{SO}** & **D_{SA}**.

D. Implementation in assistive robotics

The method is implemented on the wheelchair-based mobile manipulator EDAN in Fig. 1, for people with severe motor impairments performing activities of daily living via shared control [6], [7].

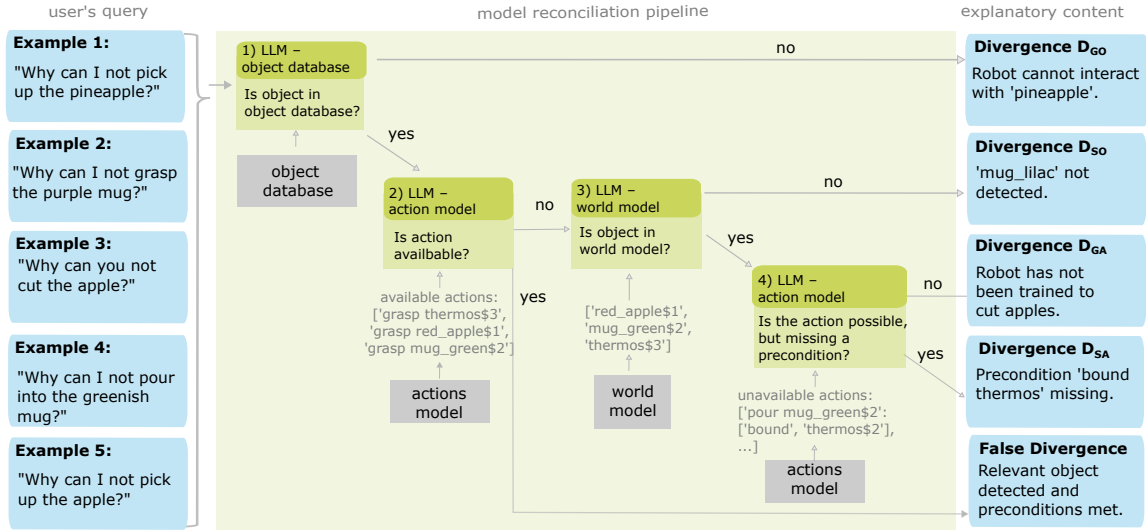


Fig. 3. Overview of the explanation generation pipeline. We note that the text in the right column refers to the explanation content and not the final sentence uttered by the LLM. See Experiments and attached video for end-to-end examples.

a) Models: Our robot represents a static object database (using a framework from [26]) with a list of object classes it knows (e.g. `green_cup`), and a world model with a list of object instances it currently locates, using perception (e.g. [`green_cup$41`, `green_cup$95`]).

For an action model we use a so-called action graph [27] that filters actions that are possible for the robot based on the objects in the world model and preconditions in the PDDL language [28]. Actions can be either *available* (preconditions are fulfilled, can be executed right away) or *blocked* (preconditions are missing). Blocked actions and their associated unmet preconditions are stored in a dictionary (see Fig. 4).

b) Explanation procedure via model matching: We implement the explanation as four search steps following the flowchart in Fig. 3. Each leaf corresponds to a divergence type (D_{GO} , D_{SO} , D_{GA} , D_{SA} , FD); the figure gives one example per type.

Step 1 - Query object database (for Divergence D_{GO}): If an object is mentioned in the human’s query, it is first verified that this object is part of the robot’s object database. The LLM workflow extracts objects from the query and attempts to match their *similarity* with the objects in the database, see prompt in Fig. 5, and provides an explanation if it fails. However, if the object is matched with an object in the database, this object name is saved and used in subsequent search steps. *Example*— The human attempts to grasp a pineapple using the robotic arm. However, this is an

unknown object to our robot. The method therefore informs the human about the inability to grasp the pineapple (see Ex. 1 in Fig. 3).

Step 2 - Query action model, first pass (for Divergence D_{SO}): It is assessed if the action (with an optional object) in the human’s query can actually be executed by the robot, by searching in the action graph’s available actions. This prevents the method from further searching, and requires the human to be more precise if the problem persists. *Example*— In Ex. 5 of Fig. 3, the human asks why an apple grasping task is not possible. As the preconditions of this task are met according to the action graph, the robot will simply explain that the grasp is possible after all given the robot models.

Step 3 - Query world model (for Divergence D_{GA}): If an object was mentioned in the human’s query, the LLM ascertains if an instance of the object name (from Step 1) is part of the robot’s world model. If it is not, it returns an explanation. *Example*— In Ex. 2 in Fig. 3 the world model contains a `red_apple$1`, `mug_green$2` and `thermos$3`, but no object `mug_lilac$id` as identified from the user query about a “purple mug”.

Step 4: Query action model, second pass (for Divergences D_{GA}/D_{SA}): The LLM searches in the blocked action dictionaries from the action graph, aiming to find if

```

A
# Dictionary containing the disabled actions as values in lists
# and the respective unmet preconditions as keys
{'bind_mug_green$2_right_arm': [['release_sct.yml', 'mug_green$2', 'right_arm']],
 'not_closed_op_microwaves$1': [['close_microwave_sct.yml', 'op_microwaves$1', 'right_arm']]}

B
# Dictionary containing the disabled actions as values in lists
# and the respective unmet preconditions as keys
{'bind_mug_green$2_right_arm': [['release_sct.yml', 'mug_green$2', 'right_arm']],
 'closed_op_microwaves$1': [['open_microwave_sct.yml', 'op_microwaves$1', 'right_arm']]}
  
```

Fig. 4. Symbolic state dictionaries for a world model with (A) a green mug and a closed microwave; (B) a green mug and an open microwave.

```

# prompt object database
Your job is to state if a target object is part of a list of objects or not. You need to assign it in one of the categories 'no' or 'yes'.
1) Write 'Thought': think step by step how to solve it. Check if there are objects in a list of objects that match the given one. If there are matches return the given object in the 'yes' category. If there are no matches return the given object in the 'no' category.
2) Write 'Final answer': return a json with two list elements 'no' and 'yes' in which you categorise the given object. Always write 'Final answer'!
Example:
- target object: ["blueish vase"]
- list: [{"vase_dark_blue", "yellow_banana"}]
- Thought: The object is "blue-ish vase". A possible match is "vase_dark_blue". I will assign "blue-ish vase" in the "yes" category.
- Final answer: {"no": [], "yes": ["blue-ish vase"]}

(Four more examples)

Actual Task:
- target object: (object_mentioned)
- list: (odb_list)
  
```

Fig. 5. Example prompt for matching an object with the robot database.

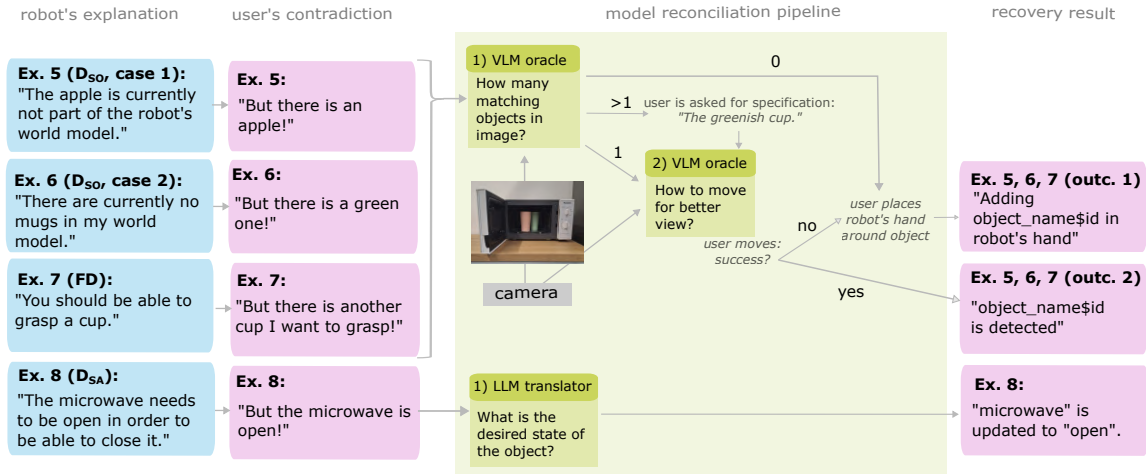


Fig. 6. Overview of the recovery workflow.

the requested action is blocked by a precondition, or not. If it is blocked it returns the missing precondition as an explanation (Divergence D_{SA}); if it is not, we assume the robot is not trained to execute the said action, and return this as an explanation (Divergence D_{GA}). *Example for D_{GA}* — The user asks in Ex. 3 about cutting an apple; despite a `red_apple$1` being located, the action cutting is not available nor blocked, so the robot explains the robot has not been trained for it. *Example for D_{SA}* — The user asks in Ex. 4 about pouring into a green mug. Intuitively, grasping a container with liquid is a precondition for this action, but in this case this is missing (e.g., the robot could have failed to detect that a thermos is grasped). As an explanation, the LLM provides the missing precondition from the action graph.

c) Recovery procedure: We implemented a recovery pipeline as a flowchart in Fig. 6. There are two cases:

Case 1: Divergences D_{SO} & FD: An object desired by the user is not in the world model (even if the explanation incorrectly assumed a False Divergence, see Ex. 7 in Fig. 6). We use a VLM as an “oracle” to (1) attempt to match the object in the user rebuttal with the objects in the camera image of the robot, shown in Ex. 6 (successful match) and Ex. 5 (unsuccessful match); and (2) if matched, provide a movement suggestion of the robot wheelchair that would achieve a better object view without occlusions, so that the robot perception algorithms can detect it. *Outcome 1:* If there is no match or the wheelchair movement fails to fix the problem, the LLM adds the missing object to the world model, asking the user to drive the end-effector toward the object to obtain a pose. *Outcome 2:* If the suggestion works, the robot’s perception algorithm automatically detects and adds the object to the world model.

Case 2: Divergence D_{SA} : In case of an unmet precondition, the desired state is collected from the human rebuttal (e.g. the open status of the microwave from “But the microwave is open!”). This is matched to a symbolic state (in PDDL), the world model (and the derived action graph) and overwritten accordingly (Ex. 8 in Fig. 6).

d) LLM/VLM: We used an open-weights local VLM for all steps, *Mistral-Small-3.2-24B-Instruct-2506* [29]. Prompt engineering techniques such as *Chain of Thought* and *Few Shot Prompting* were applied [30], [31], in order to improve the generation capabilities of the relatively-small model.

IV. EVALUATION

A. Pilot study on real robot

Do explanations and recovery work on a realistic setting?

Robot: Experiments were conducted on the wheelchair-based mobile manipulator EDAN with a lightweight arm and a gripper end-effector, shown in Figs. 7 and 8 [6], [7]. The robot end-effector was controlled with a 3D joystick in shared control. The robot offered the user a set of actions (e.g. a grasp), that could be activated by users by driving the end-effector nearby the object, and then completed using Shared Control Templates [23]. The language model was running on a workstation next to the robot (dual NVIDIA RTX 6000 Ada 48GB GPU), and we interacted with the robot via speech.

Tasks and robot models: We considered two sequences where the experimenters commanded the robot to interact with daily living objects, shown in Fig. 7 and Fig. 8 respectively, and added to the video submission. The robot located known objects via a perception pipeline and updated its world and action models accordingly. Known objects were a thermos bottle, a mug, and an apple, while unknown objects (not in the database) were two toys (pineapple, octopus).

Explanation and recovery: On each of the user queries on Figs. 7 and 8 the robot did not offer to activate in shared control the tasks that users were asking about (except for divergence FD on Frame 4 of Fig. 7). The first sequence provides examples of the explanation procedure only, highlighting one query of each divergence type. The second sequence shows the full explanation and recovery process from Section III-Dc, but removing the VLM oracle step with movement suggestions for simplicity of example (featured later on Section IV-B). After the recovery interactions on the






					
World model contents	[drawer handle id 653, drawer handle id 654]	Same as Frame 1 + green apple id 659	Same as Frame 2	Same as Frame 2 + peach mug id 661	Same as Frame 4 + thermos blue id 664
User query	"Why can you not pick up the pineapple?"	"Why can you not cut the apple?"	"Why can you not pick up the mug?"	"Why can you not pick up the mug now?"	"Why can you not pick up the thermos?"
Robot reply	"I am afraid the pineapple is not part of my object database, so you cannot perform an action with it" [Divergence D_{co}]	"Sorry, but cutting an apple is not something I can do" [Divergence D_{ca}]	"I am afraid I can currently not pick up because the mug is not part of my world representation" [Divergence D_{so}]	"You should be able to pick up the mug based on current model of the world" [Divergence FD]	"You need to ensure the robot gripper is free in order to (...) pick up an object" [Divergence D_{ca}]

Fig. 7. First video sequence. Explanation results with a real robot with several objects. The user guides the robot to pick up the mug between frames 4 and 5. We note the world model also represents the user and the end-effector. The full sequence is in the attached video.




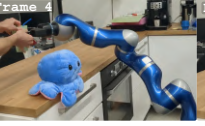


						
	Problem 1: object (plushie) prevents robot from locating and grasping a green_mug object.			Problem 2: the mug was removed from the robot end-effector, unbeknown to the robot models		
Expl.	User: "Why can I not grasp the greenish cup?"	Robot: "(...) the greenish cup is not part of my world representation"		User: "Why can I not grasp the thermos bottle?"	Robot: "you need to ensure that the robot gripper is free (...)"	
Rec.	User: "There is a green cup there!"	Robot [after world model update]: "Thank you, I have corrected the issue"		User: "Right now the gripper is free!"	Robot [after world model update]: "Thank you, I have corrected the issue"	

Fig. 8. Second video sequence. Recovery results with a real robot with a thermos and a mug. The full sequence is in the attached video.

second sequence, the tasks that were blocked (“grasping the greenish cup”, “grasping the thermos bottle”) were offered again to the user as the issues were solved, and executed in shared control (frames 3 and 6 in Fig. 8).

B. Evaluation with a robot simulation

We conducted two experiments on a digital twin, **Experiment B1** and **Experiment B2**. Exp. B1 used three unit tests: two for generating correct explanations for world and action divergences (D_{SO} , D_{SA}), and one for recovery suggestions for a failed object localization (D_{SO}). Exp. B2 revisits the action divergence unit test to evaluate the effect of robot semantic representations.

Dataset: A dataset composed of 40 episodes was gathered. Each episode represents one human-robot-interaction for one of the units. There are 13 units for failed object localization, 11 for unmet predicates and 16 for recovery suggestions. An episode consists of an image, a matching world model of the robot, and an initial human’s query, plus potential follow-up contradictions or specifications from the human. The images were partially collected from our robot’s camera itself in the kitchen of the lab. Additional image material was gathered in the kitchen and the living room of the lab as well as our office using a smartphone. All images contain at least one object from our robot’s object database. An example of an episode can be seen in Fig. 9. For each episode a ground truth conversation was created. This conversation contains the expected explanations or recovery suggestions based on


	World Model	Ground Truth	Method Output	Naive VLM Output
	"mug_peach" (in image, free), "op_microwave" (in image, in world, closed) "edan_hand" (free)	{"user": "Why can I not close the microwave?"}, {"robot": "You need to open the microwave first to meet the necessary precondition to close."}	{"user": "Why can I not close the microwave?"}, {"robot": "You need to open the microwave first to meet the necessary precondition to close."}	{"user": "Why can I not close the microwave?"}, {"robot": "I cannot close the microwave because it contains items that need to be removed first, such as the two cups currently placed on the microwave's interior surface."}

Fig. 9. Example episode for the unit testing explanation generation for unmet task precondition

the given human’s query, image, and world model.

Answer generation and labeling: For all experiments (Exp. B1 and B2), explanations and recovery suggestions were computed from a digital twin containing the robot’s world and action models. As each episode was run three times, we have 120 output conversations in total.

We assigned it a label of True if the method reply corresponded to the ground truth explanation in the dataset using the criteria in Fig. 10. We obtained two types of labels: (1) the label by one of the authors; and (2) the label by *Mistral-Small-3.2-24B-Instruct-2506* [29] as in “LLM-as-a-judge” [32] (prompted with the label criteria displayed in

unit tested	criteria for “true” label
explanation object not in world	correct explanation of object(s) mentioned or needed for task performance being in the world or not.
explanation unmet precondition	correct explanation of the preconditions needed for task performance
recovery suggestion for object not in world	correct movement suggestion for the correct object or statement of the object not being in the environment.

Fig. 10. Evaluation rating criteria

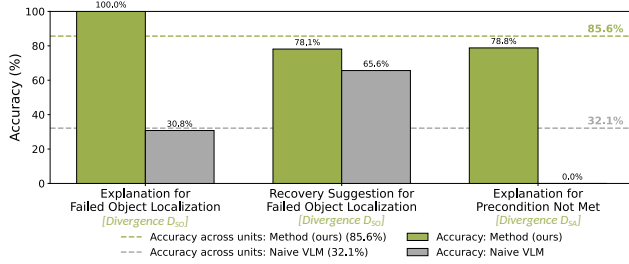


Fig. 11. Accuracy scores for each unit for our method and the naive VLM

Fig. 10). The goal of using an LLM to label the data is to ensure unbiased labeling if only an author would label the data. In all figures in this subsection, we report accuracy as the *average* between LLM and human labels. We obtained a satisfactory inter-rater reliability, i.e. *Cohen's* $\kappa = 0.91$.

a) *Experiment B1: ablation of our framework: Question: Are the explanations common-sense, or is the robot's model required for correctly explaining and suggesting recovery strategies? Benchmark with a Naive VLM:* To draw a comparison, we ablated our framework by directly querying a “naive” VLM (again *Mistral-Small-3.2*) with the same dataset including the image of the episodes. We used two prompts (one for the two explanation units in Fig. 12 and one for the recovery suggestion unit), where general context about our robot, the task and the human's query was provided. However, the prompts did not contain information from robot's models. **Results:** The accuracy was calculated for each unit separately, as shown in Fig. 11. Examples of method and naive VLM replies are shown in Fig. 9.

b) *Experiment B2: semantic representations: Question: What is our reliance on semantically rich representations?*

By inspecting failed episodes of the unit for unmet preconditions, it seems that failures are mainly due to misinterpretations of uncommon English robot representations. For instance, the cabinet of the kitchen is represented on the robot's object database as `ikea_bagganas`, the name of the product, which the LLM does not always identify as a cabinet, and sometimes mistakes it for an “ikea bag”. We therefore repeat the unit test for “Precondition not met” (D_{SA}), but augmented the relevant prompts with a “translation dictionary” of uncommon terms.

Results: As shown in Fig. 13, providing a dictionary with translation of uncommon terms yielded an accuracy of 92.42%, 13.63% more than in Experiment B1.

```
# prompt naive vlm explanation:
You are the speech AI module of an assistive mobile wheelchair robot. Your goal is to provide an explanation to the user about why a certain task is currently not executable. You must adapt the persona of the robot.
To answer:
* Provide an explanation to the user's query about a certain task not being executable.
* This fault behavior of the robot is due to one of the following: 1) the user's target object is not part of the robot's world representation 2) the symbolic state of the world was misinterpreted by the robot which is why preconditions for task execution are not met. State which precondition is unmet.
* If you think an object should have been successfully detected and included to the robot's world representation state that as well.
* The photo attached comes from the robot's camera, representing what the robot "sees".
* Write "Thought:" and very briefly think your answer step by step.
* Write "Final answer:" and provide your explanation as if it were given by the robot. Use first person, and impersonate the robot.
* Use the photo for additional information to provide a good explanation, but do not use the statement "in the photo provided". Do not speak about "the attached image".
Remember, this camera image contains everything EDAM "sees".
The user's question is: (users_query)
```

Fig. 12. Naive VLM prompt for explanations

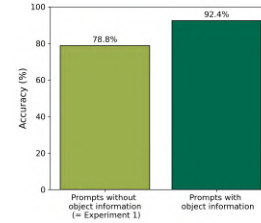


Fig. 13. Accuracy for the explanation for precondition not met unit for prompts without information about robotic object representation vs. prompts with information about it

C. Discussion

As shown in our robot experiments (Figs. 7 and 8) and validated in our simulation unit tests (Figs. 11 and 13), our method can create explanations and recovery suggestions for a variety of daily living objects and task situations, answering all real robot queries correctly on a long sequence and achieving between 78 and 100% accuracy on the diverse unit tests. Experiment B1 shows that the model reconciliation outperforms a naive vision-only baseline, achieving 100% accuracy in explaining object localization failures, 78.79% in communicating symbolic state errors, and 78.12% in recovery suggestions. Experiment B2 showed that adding a dictionary with definitions of uncommon terms (such as in the `ikea_bagganas` kitchen cabinet example) further increases the accuracy from 78.8 to 92.4% to the action reconciliation experiment, emphasizing the requirement of having clear robot representations that can be interpreted in English by the LLM. Furthermore, a naive VLM without the model reconciliation context has considerably lower accuracy in all conducted tests, which signifies that the LLM retrieves and provides the correct information from the modules of the robot to explain a failure situation, and a VLM by itself is not able to produce those explanations.

Our evaluation reveals several limitations that point to future improvements. As shown in the unit test in Fig. 11, recovery movement suggestions could be better adapted to our robot platform; the accuracy drop appears to stem from incorrect platform suggestions by the VLM, likely reflecting spatial reasoning limitations of smaller models [33]. We also observed that small LLMs can be sensitive to adjectives, occasionally inferring objects (e.g., a “greenish mug”) from loosely related cues such as the presence of a green apple, highlighting brittleness in semantic grounding.

In addition, our experiments cover only a limited range of everyday scenarios, and broader evaluations with more diverse tasks and objects are needed to better assess transparency and usability. The specific contribution of LLM-driven semantics should be isolated using stronger baselines, including a purely symbolic PDDL/action-graph approach with template-based explanations and a classical plan-explanation method. Finally, user-driven state overwrites require more systematic safety validation, for instance through cross-checks with perception confidence.

To resolve these limitations, in particular, a future user study would be appropriate, testing if the application of the method is indeed perceived as helpful and leads to more trust

and improved human-robot-teaming.

V. CONCLUSION

We introduced a model-reconciliation framework using foundation models to explain and recover from unexpected robot behavior. Experiments show it provides accurate explanations and reasonable recovery suggestions, given that robot models contain English-interpretable representations.

In future work, we will explore explanations related to *plan optimality* (explaining why a plan is optimal with longer planning horizons), *goals* (resolving mismatched goals between human and robot), and *failed feasibility checks* (resolving situations where an action is possible in theory, but not in practice [34]). We will also support multiple instances of the same object type, such as multiple drawers requiring the human to specify disambiguating properties, such as “why cannot I grasp the *second drawer from the top*”. Finally, we will consider comparisons with other interface modalities, as visual explanations of a robotic wheelchair’s model could improve predictability, recovery times, and understanding of the robot’s inner workings [35], [36].

REFERENCES

- [1] S. Wallkötter, S. Tulli *et al.*, “Explainable Embodied Agents Through Social Cues: A Review,” *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 10, no. 3, pp. 1–24, 2021.
- [2] A. Tabrez, M. B. Luebbbers, and B. Hayes, “A survey of mental modeling techniques in human-robot teaming,” *Current Robotics Reports*, vol. 1, pp. 259–267, 2020.
- [3] M. Zakershaharak, S. R. Marpally *et al.*, “Order matters: Generating progressive explanations for planning tasks in human-robot teaming,” in *2021 ICRA*. IEEE, 2021, Conference Proceedings, pp. 3751–3757.
- [4] T. Chakraborti, S. Sreedharan *et al.*, “Plan explanations as model reconciliation: Moving beyond explanation as soliloquy,” in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 156–163.
- [5] E. Schluntz and B. Zhang. (2024) Building effective agents. Engineering at Anthropic blog. [Online]. Available: <https://www.anthropic.com/engineering/building-effective-agents>
- [6] J. Vogel, A. Hagenhuber *et al.*, “EDAN: An EMG-controlled daily assistant to help people with physical disabilities,” in *2020 IROS*. IEEE, 2020, Conference Proceedings, pp. 4183–4190.
- [7] J. Vogel, M. Atzenhofer *et al.*, “Shared control in assistive robotics: A cybathlon-winning approach,” *Science Robotics*, vol. 10, no. 107, p. eae6725, 2025.
- [8] T. Miller, “Explanation in artificial intelligence: Insights from the social sciences,” *Artificial intelligence*, vol. 267, pp. 1–38, 2019. [Online]. Available: <https://doi.org/10.1016/j.artint.2018.07.007>
- [9] G. A. Radvansky and J. M. Zacks, “Event perception,” *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 2, no. 6, pp. 608–620, 2011.
- [10] H. Wimmer and J. Perner, “Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children’s understanding of deception,” *Cognition*, vol. 13, no. 1, pp. 103–128, 1983.
- [11] H. Dung and T. Son, “On model reconciliation: How to reconcile when robot does not know human’s model?” *Electronic Proceedings in Theoretical Computer Science*, vol. 364, pp. 27–48, 08 2022.
- [12] S. L. Vasileiou, A. Previti, and W. Yeoh, “On exploiting hitting sets for model reconciliation,” in *AAAI Conference on Artificial Intelligence (AAAI-21)*, 2023.
- [13] A. Hagenhuber, G. Quere *et al.*, “An assistive robot that enables people with amyotrophy to perform sequences of everyday activities,” *Scientific Reports*, vol. 15, no. 1, p. 8426, 2025.
- [14] N. Moorman, M. Zhao *et al.*, “Bi-directional mental model reconciliation for human-robot interaction with large language models,” in *Workshop in Advancing Artificial Intelligence through Theory of Mind, AAAI*, 2025.
- [15] R. Wu, S. Kortik, and C. H. Santos, “Automated behavior tree error recovery framework for robotic systems,” in *2021 ICRA*. IEEE, 2021, Conference Proceedings, pp. 6898–6904.
- [16] J. Blankenburg, M. Zagainova *et al.*, “Human-robot collaboration and dialogue for fault recovery on hierarchical tasks,” in *Social robotics: Proceedings of the 12th International Conference*, A. R. Wagner, D. Feil-Seifer *et al.*, Eds. Springer, 2020, pp. 144–156.
- [17] S. Klein, J. Huch *et al.*, “Creating a framework for a user-friendly cobot failure management in human-robot collaboration,” in *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, 2024, Conference Proceedings, pp. 618–622.
- [18] T. Eiband, C. Willibald *et al.*, “Collaborative programming of robotic task decisions and recovery behaviors,” *Autonomous Robots*, vol. 47, no. 2, pp. 229–247, 2023.
- [19] G. Chance, A. Camilleri *et al.*, “An assistive robot to support dressing-strategies for planning and error handling,” in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechanics (BioRob)*. IEEE, 2016, Conference Proceedings, pp. 774–780.
- [20] K. Hori, K. Suzuki, and T. Ogata, “Interactively robot action planning with uncertainty analysis and active questioning by large language model,” in *2024 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2024, Conference Proceedings, pp. 85–91.
- [21] G. Sarch, Y. Wu *et al.*, “Open-ended instructable embodied agents with memory-augmented large language models,” *PsyArXiv*, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2310.15127>
- [22] L. Zha, Y. Cui *et al.*, “Distilling and retrieving generalizable knowledge for robot manipulation via language corrections,” in *2024 ICRA*. IEEE, 2024, pp. 15 172–15 179.
- [23] G. Quere, A. Hagenhuber *et al.*, “Shared control templates for assistive robotics,” in *2020 ICRA*, 2020, Conference Proceedings, pp. 1956–1962.
- [24] R. Sakagami, F. S. Lay *et al.*, “Robotic world models – conceptualization, review, and engineering best practices,” *Frontiers in Robotics and AI*, vol. 10, November 2023.
- [25] S. Bustamante, G. Quere *et al.*, “CATs: Task planning for shared control of assistive robots with variable autonomy,” in *2022 ICRA*. IEEE, 2022, Conference Proceedings, pp. 3775–3782.
- [26] D. S. Leidner, *Cognitive reasoning for compliant robot manipulation*. Springer, 2019, vol. 23.
- [27] M. Behery, “A knowledge-based activity representation for shared autonomy teleoperation of robotic arms,” RWTH-Aachen University, Master’s Thesis, 2016.
- [28] M. Ghallab, C. Knoblock *et al.*, “PDDL - The Planning Domain Definition Language,” 1998. [Online]. Available: https://planning.wiki/_citedpapers/pddl1998.pdf
- [29] Mistral AI, “Mistral-small-3.2-24b-instruct-2506,” <https://huggingface.co/mistralai/Mistral-Small-3.2-24B-Instruct-2506>, Jul. 2025, a minor update over Small-3.1, improved instruction following, reduced repetition, and more robust function calling.
- [30] J. Wei, X. Wang *et al.*, *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. Curran Associates, Inc., 2022, vol. 35, pp. 24 824–24 837.
- [31] T. Gao, A. Fisch, and D. Chen, “Making pre-trained language models better few-shot learners,” in *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, 2021, pp. 3816–3830.
- [32] J. Gu, X. Jiang *et al.*, “A survey on llm-as-a-judge,” 2025. [Online]. Available: <https://arxiv.org/abs/2411.15594>
- [33] A. Pothiraj, E. Stengel-Eskin *et al.*, “Capture: Evaluating spatial reasoning in vision language models via occluded object counting,” 2025. [Online]. Available: <https://arxiv.org/abs/2504.15485>
- [34] S. Bustamante, I. Rodríguez *et al.*, “Feasibility checking and constraint refinement for shared control in assistive robotics,” *IEEE Robotics and Automation Letters*, vol. 9, no. 9, pp. 8019–8026, 2024.
- [35] M. Zolotas and Y. Demiris, “Transparent intent for explainable shared control in assistive robotics,” in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 2021, Conference Proceedings, pp. 5184–5185.
- [36] C. Brooks and D. Szafrir, “Visualization of intended assistance for acceptance of shared control,” in *2020 IROS*. IEEE, 2020, pp. 11 425–11 430.