Generating Explanations of Action Failures in a Cognitive Robotic Architecture

Ravenna Thielstrom, Antonio Roque, Meia Chita-Tegmark, Matthias Scheutz
Human-Robot Interaction Laboratory
Tufts University
Medford, MA 02155
{ravenna.thielstrom,antonio.roque,mihaela.chita_tegmark,matthias.scheutz}@tufts.edu

Abstract

We describe an approach to generating explanations about why robot actions fail, focusing on the considerations of robots that are run by cognitive robotic architectures. We define a set of Failure Types and Explanation Templates, motivating them by the needs and constraints of cognitive architectures that use action scripts and interpretable belief states, and describe content realization and surface realization in this context. We then describe an evaluation that can be extended to further study the effects of varying the explanation templates.

1 Introduction

Robots that can *explain* why their behavior deviates from user expectations will likely benefit by better retaining human trust (Correia et al., 2018; Wang et al., 2016). Robots that are driven by a cognitive architecture such as SOAR (Laird, 2012), ACT-R (Ritter et al., 2019), or DIARC (Scheutz et al., 2019) have additional requirements in terms of connecting to the architecture's representations such as its belief structures and action scripts. If properly designed, these robots can build on the interpretability of such architectures to produce explanations of action failures.

There are various types of cognitive architectures, which may be defined as "abstract models of cognition in natural and artificial agents and the software instantiations of such models" (Lieto et al., 2018) but in this effort we focus on the type that uses action scripts, belief states, and natural language to interact with humans as embodied robots in a situated environment. In Section 2 we describe an approach to explaining **action failures**, in which a person gives a command to a robot but the robot is unable to complete the action. This approach was implemented in a physical robot with a cognitive architecture, and tested with a preliminary evaluation as described in Section 3. After comparing our effort to related work in Section 4, we finish by discussing future work.

2 An Approach to Action Failure Explanation

Our approach is made up of a set of Failure Types, a set of Explanation Templates, algorithms for Content Realization, and algorithms for Surface Realization.

2.1 Failure Types

We have defined an initial set of four different failure types, which are defined by features that are relevant to cognitive robots in a situated environment. One approach to designing such robots is to provide a database of action scripts that it knows how to perform, or that it is being taught how to perform. These scripts often have prerequisites that must be met before the action can be performed; for example, that required objects must be available and ready for use. These action scripts also often have defined error types that may occur while the action is being executed, due to the unpredictability of the real world. Finally, in open-world environments robots usually have knowledge about whether a given person is authorized to command a particular action. Incorporating these feature checks into the architecture of the robot allows for automatic error type retrieval when any of the checks fail, essentially providing a safety net of built-in error explanation whenever something goes wrong. These features are used to define the failure types as follows. When a robot is given a command, a series of checks are performed.

First, for every action necessary to carry out that command, the robot checks to see whether the action exists as an action script in the robot's database of known actions. If it does not, then the action is not performed due to an **Action Ignorance** failure type. This would occur in any situation where the robot lacks knowledge of *how* to perform an action, for example, if a robot is told to walk in a circle, but has not been instructed what walking in a circle means in terms of actions required.

Second, the robot checks whether it is obligated to perform the action, given its beliefs about the authorization level of the person giving the command. If the robot is not obligated to perform the action, the system aborts the action with an **Obligation Failure** type. An example of this failure would be if the person speaking to the robot does not have security clearance to send the robot into certain areas.

Third, the robot checks the conditions listed at the start of the action script, which define the facts of the environment which must be true before the robot can proceed. The robot evaluates their truth values, and if any are false, the system exits the action with a **Condition Failure** type. For example, a robot should check prior to walking forward that there are no obstacles in its way before attempting that action.

Otherwise, the robot proceeds with the rest of the action script. However, if at any point the robot suffers an internal error which prevents further progress through the action script, the system exits the action with a **Execution Failure** type. These failures, in contrast, to the pre-action condition failures, come during the execution of a primitive action. For example, if a robot has determined that it is safe to walk forward, but after engaging its motors to do just that, either an internal fault with the motors or some other unforseen environmental hazard result in the motors not successfully engaging. In either case, from the robot's perspective, the only information it has is that despite executing a specific primitive (engaging the motors), it did not successfully return the expected result (motors being engaged).

2.2 Explanation Templates

Once the type of failure is identified, the explanation assembly begins. The basic structure of the explanation is guided by the nature of action scripts. We consider an inherently interpretable action representation that has an intended goal **G** and failure reason **R** for action **A**, and use these to build four different explanation templates of varying depth.

The **GA** template captures the simplest type of explanation: "I cannot achieve *G* because I cannot do *A*." For example, "I cannot *prepare the product* because I cannot *weigh the product*."

The **GR** template captures a variant of the first explanation making expicit reference to a reason: "I cannot achieve *G* because of *R*." For example, "I cannot *prepare the product* because *the scale is occupied*."

The **GGAR** template combines the above two schemes by explicitly linking *G* with *A* and *R*: "I cannot achieve *G* because to achieve *G* I must do *A*, but *R* is the case." For example, "I cannot *prepare the product* because to *prepare something* I must *weigh it*, but *the scale is occupied*."

Finally, the **GGAAR** template explicitly states the goal-action and action-failure reason connections: "I cannot achieve *G* because for me to achieve *G* I must do *A*, and I cannot do *A* because of *R*." For example, "I cannot *prepare the product* because to *prepare something* I must *weigh it*, and I cannot *weigh the product* because *the scale is occupied*."

2.3 Content Realization

Given the failure type that has occurred, and the explanation template (which is either set as a parameter at launch-time or determined at run-time), a data structure carrying relevant grammatical and semantic information is constructed.

The code version of an explanation template contains both bound and generic variables, which in the GGAAR template looks like:

```
can(not(BOUND-G),
    because(advrb(infinitive(GENERIC-G),
    must(GENERIC-A)), can(not(BOUND-A),
    because(REASON))))
```

`BOUND-G` and `GENERIC-G` are the bound and unbound versions of the goal. For example `did(self,prepare(theProduct))` is the bound version which specifies the product, and `did(self,prepare(X))` is the unbound version.

Similarly, `GENERIC-A` is the generic form of the sub-action which failed, such as `did(self,weigh(X))`, `BOUND-A` is the lowest-level sub-action, such as `did(self, weigh(theProduct))`, and `REASON` is the error reason, such as `is(theScale,occupied)`.

So the resulting form would look like:

```
can(not(prepare(self,theProduct)),
    because(advrb(infinitive(
    prepare(self,X)), must(
    weigh(self,X))),
    can(not(weigh(self,theProduct)),
    because(is(theScale,occupied)))))
```

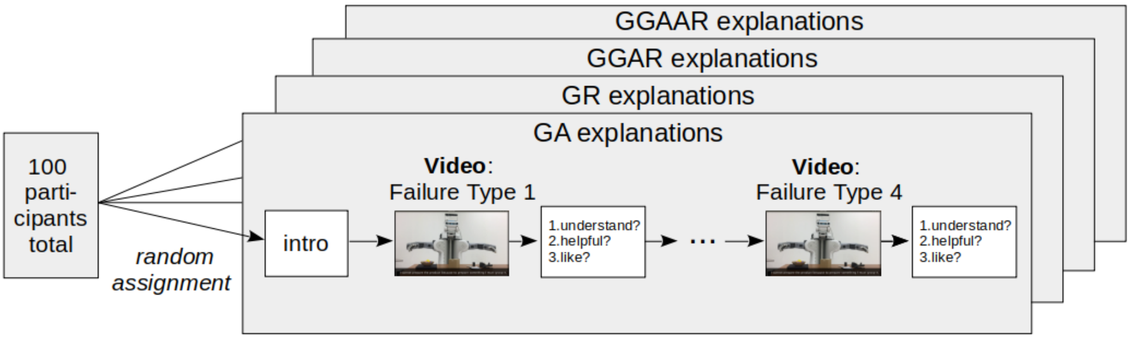


Figure 1: Study Procedure.

and would then be submitted to the Surface Realization process.

2.4 Surface Realization

Translating the semantic form of the explanation into natural language is a matter of identifying grammatical structures such as premodifiers, infinitives, conjunctions, and other parts of speech by recursively iterating through the predicate in search of grammar signifiers.

This process involves populating grammatical data structures (i.e. clauses) with portions of the semantic expression and their relevant grammatical information. During each recursive call, the name of the current term is checked to see if it matches a grammatical signifier; if so, it is unwrapped further and recurses over the inner arguments. Without any more specific signifiers, the term name can be assumed to be a verb, the first argument the subject, and the second the object of the clause. The grammatical signifiers are used to assign grammatical structure as needed, which are then conjugated and fully realized using SimpleNLG (Gatt and Reiter, 2009) into natural language, such as: "I cannot prepare the product because to prepare something I must weigh it, and I cannot weigh the product because the scale is occupied."

3 Evaluation

To validate our system, we conducted a user study. Besides testing the components all working together, we were also interested in understanding the effect of the different types of explanation templates on human perceptions of the explanations given. This study was conducted under the oversight of an Institutional Review Board.

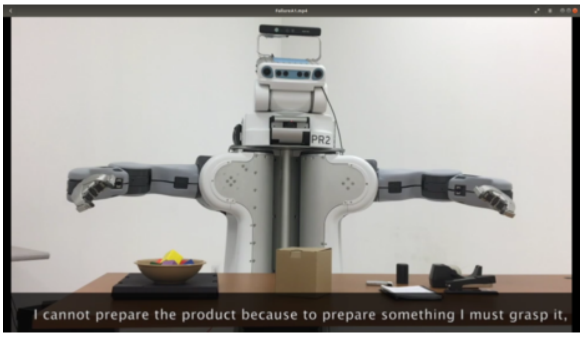
3.1 Methods

100 participants were recruited via Amazon's Mechanical Turk and completed this study online through a web interface.

As shown in Figure 1, after a brief introduction, participants were shown four different videos, one at a time, in which a robot was instructed to "prepare the product." In each video the robot explained that it could not complete the task due to one of four failure types described in Section 2.1. For example, in the first video the robot might explain that it did not know how to perform the action, in the second video the robot might explain that the person was not authorized to make the action request, in the third video the robot might explain that the scale was occupied, and in the fourth video the robot might explain that their pathfinding algorithm had failed. 25 participants were shown videos in which the explanations used the GA template, 25 in which the videos used the GR template, 25 with the GGAR template, and 25 with the GGAAR template.

After each video the participants were asked three questions.

First, to assess their understanding of how the robot failed its task, the participants were asked "What would you do in order to allow the robot to complete the task?" and were given 5 possible solutions in a multiple-choice format, only one of which was correct. For example, given the Condition Failure error explanation in the GGAAR format: "I cannot prepare the product because to prepare the product I must weigh it, and I cannot weigh the product because the scale is occupied" possible solutions are: (1) I would have the robot learn how to weigh things, (2) I would have the robot's pathfinding component debugged, (3) I would clear the scale, (4) I would move the scale closer to the



I cannot prepare the product because to prepare something I must grasp it,

Figure 2: Screen Capture from Example Video with Generated Text. A robot, given an instruction, explains an action failure.

robot, (5) I would have the robot's vision sensors repaired, where 3 is the correct solution.

Second, the participants were asked "How helpful was the robot's explanation?" on a 5-point Likert scale where 1 was "Not at all" and 5 was "Extremely."

Third, the participants were asked "How much did you like the robot's explanation?" on a 5-point Likert scale where 1 was "Not at all" and 5 was "Extremely."

These questionnaire items were selected with a focus on the social interaction between the robot and the human rather than the fluency or semantic meaning of the natural language generation itself. Perceived helpfulness and likability are both metrics of trust in a human-robot interaction, and more specifically, they are indications of the human being comfortable cooperating with the robot. Thus we aimed to assess how well the robot's explanation communicated the problem to the human (with the accuracy questions), in addition to how successful the explanations were as a social interaction.

The failure explanations in the videos were generated using a Wizard-of-Oz approach. Our explanation approach was implemented in a PR2 robot using the DIARC cognitive architecture (Scheutz et al., 2019). We filmed a PR2 robot performing preparatory-type movement (looking down at a table full of miscellaneous items, raising its hands, looking back up at the camera) before halting and delivering an audio failure explanation report (generated by our system as described in Section 2 and recorded separately, then edited into the video along with subtitles.) A screen capture of an example video is shown in Figure 2. An example video of an explanation is located here:

https://youtu.be/2j7r1S6zT90

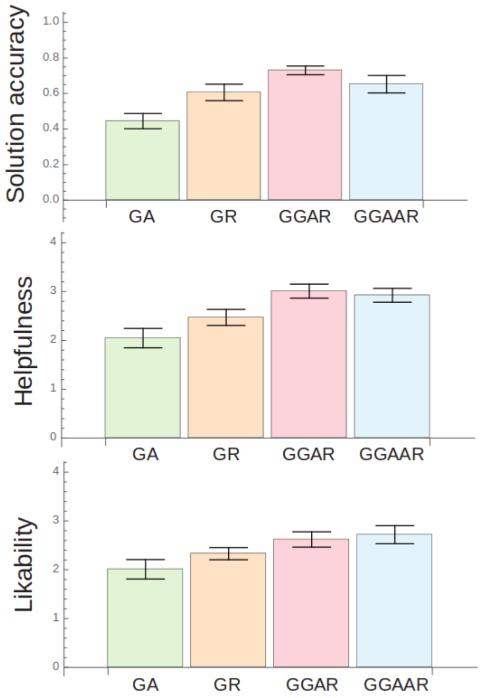


Figure 3: Evaluation Results. Proportion of accurate responses, and Likert-scale ratings of likability and helpfulness, based on Explanation Template.

3.2 Results

To investigate how the different explanation schemas the robot gave allowed the participants to select the **accurate solution** for fixing the problem, we conducted a one-way ANOVA with the solution accuracy (number of correct solutions selected across 4 different error types) as our dependent variable, and explanation template (GA, GR, GGAR and GGAAR) as the independent variable. We observed a significant effect of explanation template on solution accuracy $F(3, 97) = 8.61$, $p < .001$, η_p^2=.21. Further pairwise comparisons with Tukey-Kramer corrections revealed that GA explanations lead to significantly lower solution accuracy than GGAAR ($p = .004$), GAR ($p < .001$) and GR ($p = .031$) explanations. No other significant differences between explanation templates were observed. In other words, short explanations lacking a reason for failure will result in decreased understanding of how to best address the failure.

We then studied perceived **explanation helpfulness**. We conducted a one-way ANOVA with explanation helpfulness as the dependent variable and explanation template (GA, GR, GGAR, GGAAR) as the independent variable. We found a significant effect of explanation template, $F(3, 97) = 7.34$, $p < .001$, η_p^2=.30. Pairwise comparisons revealed a similar pattern of results as for solution accuracy: participants perceived the GA explanations to be less helpful than GGAAR ($p = .002$) and GGAR ($p < .001$), however, unlike the solution accuracy no significant differences were found between GA-type explanations and GR-type ones. No other significant differences in helpfulness were found between explanation.

Finally, we investigated **explanation likability** by conducting a one-way ANOVA with explanation likability as the dependent variable and explanation template (GA, GR, GGAR, GGAAR) as the independent variable. We found again a significant main effect of explanation schema $F(3, 96) = 3.59$, $p = .016$, η_p^2=.10. Pairwise comparisons revealed that GA explanations were liked less than GGAAR ($p = 0.021$) and GGAR ($p = 0.053$) but not significantly different from GR. We found no other significant differences in perceived likability between explanation templates.

This study highlights the value of providing a failure reason R in the explanation templates, which is shown by the reduced measures of the GA explanations.

4 Related Work

Human-Robot Interaction (HRI) research on explaining the actions of robots (Anjomshoae et al., 2019) is related to research on explaining planning decisions (Fox et al., 2017; Krarup et al., 2019), on generating language that describes the pre- and post-conditions of actions in planners (Kutlak and van Deemter, 2015), and on generating natural language explanations from various types of meaning representations (Horacek, 2007; Pourdamghani et al., 2016).

In HRI work that focuses on error reporting, Briggs and Scheutz (2015) defined a set of *felicity conditions* that must hold for a robot to accept a command. They outlined an architecture that reasons about whether each felicity condition holds, and they provided example interactions, although they did not evaluate an implementation of their approach. Similarly, Raman et al. (2013) used a logic-based approach to identify whether a command can be done, and provided example situations, but no evaluation. Our approach is similar in that we define a set of failure types for action commands, but we implement and evaluate our approach with a user study. Other recent HRI work has included communicating errors using non-verbal actions to have a robot express its inability to perform an action (Kwon et al., 2018; Romat et al., 2016), which does not focus on more complex system problems using natural language communications as we do.

There has also been recent work on user modeling and tailoring responses to users in robots (Torrey et al., 2006; Kaptein et al., 2017; Sreedharan et al., 2018). In one effort worth building upon, Chiyah Garcia et al. (2018) used a human expert to develop explanations for unmanned vehicle decisions. These explanations followed Kulesza et al. (2013) in being characterized in terms of *soundness*, relating the depth of details, and *completeness*, relating to the number of details. Chiyah Garcia et al. found links between the "low soundness and high completeness" condition and intelligibility and value of explanations.

5 Conclusions

We have described an approach to generating action failure explanations in robots, focusing on the needs and strengths of a subset of cognitive robotic architectures. This approach takes advantage of the interpretability of action scripts and belief representations, and is guided by recent directions in HRI research. Importantly, the explanation of this approach is not a post-hoc interpretation of a black-box system, but is an accurate representation of the robot's operation.

Various aspects of the approach are being continually refined. Currently, new Failure Types are being investigated, and the content realization and surface realization algorithms are being revised and tested.

Finally, the evaluation in Section 2.2 describes a preliminary approach to comparing the relative impact of the various explanation templates. We are pursuing additional studies focusing on varying the explanations produced. Initial studies would be video-based, after which follow-up studies would be conducted in the context of a task being performed either in person, or via a virtual interface that we have constructed, and the goal would be to examine the ways that context features such as user model, physical setting, and task state affect the type of explanation required.

Acknowledgment

We are grateful to the anonymous reviewers for their helpful comments. This work was in part funded by ONR grant #N00014-18-2503.

References

Sule Anjomshoae, Amro Najjar, Davide Calvaresi, and Kary Främling. 2019. Explainable agents and robots: Results from a systematic literature review. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*.

Gordon Michael Briggs and Matthias Scheutz. 2015. "Sorry, I Can't Do That": Developing mechanisms to appropriately reject directives in human-robot interactions. In *2015 AAAI fall symposium series*.

Francisco Javier Chiyah Garcia, David A. Robb, Xingkun Liu, Atanas Laskov, Pedro Patron, and Helen Hastie. 2018. Explainable autonomy: A study of explanation styles for building clear mental models. In *Proceedings of the 11th International Conference on Natural Language Generation*.

Filipa Correia, Carla Guerra, Samuel Mascarenhas, Francisco S. Melo, and Ana Paiva. 2018. Exploring the impact of fault justification in human-robot trust. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18.

Maria Fox, Derek Long, and Daniele Magazzeni. 2017. Explainable planning. In *Proceedings of the IJCAI 2017 Workshop on Explainable AI*.

Albert Gatt and Ehud Reiter. 2009. SimpleNLG: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 90–93.

Helmut Horacek. 2007. How to build explanations of automated proofs: A methodology and requirements on domain representations. In *Proceedings of AAAI ExaCt: Workshop on Explanation-aware Computing*, pages 34–41.

Frank Kaptein, Joost Broekens, Koen Hindriks, and Mark Neerincx. 2017. Personalised self-explanation by robots: The role of goals versus beliefs in robot-action explanation for children and adults. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 676–682. IEEE.

Benjamin Krarup, Michael Cashmore, Daniele Magazzeni, and Tim Miller. 2019. Model-based contrastive explanations for explainable planning. In *Proceedings of the ICAPS 2019 Workshop on Explainable Planning (XAIP)*.

Todd Kulesza, Simone Stumpf, Margaret Burnett, Sherry Yang, Irwin Kwan, and Weng-Keen Wong. 2013. Too much, too little, or just right? Ways explanations impact end users' mental models. In *2013 IEEE Symposium on Visual Languages and Human Centric Computing*, pages 3–10. IEEE.

Roman Kutlak and Kees van Deemter. 2015. Generating Succinct English Text from FOL Formulae. In *Procs. of First Scottish Workshop on Data-to-Text Generation*.

Minae Kwon, Sandy H Huang, and Anca D Dragan. 2018. Expressing robot incapability. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 87–95. ACM.

John E Laird. 2012. *The Soar cognitive architecture*. MIT press.

Antonio Lieto, Mehul Bhatt, Alessandro Oltramari, and David Vernon. 2018. The role of cognitive architectures in general artificial intelligence. *Cognitive Systems Research*, 48:1 – 3.

Nima Pourdamghani, Kevin Knight, and Ulf Hermjakob. 2016. Generating English from abstract meaning representations. In *Proceedings of the 9th International Natural Language Generation conference*, pages 21–25.

Vasumathi Raman, Constantine Lignos, Cameron Finucane, Kenton CT Lee, Mitchell P Marcus, and Hadas Kress-Gazit. 2013. Sorry Dave, I'm afraid I can't do that: Explaining unachievable robot tasks using natural language. In *Robotics: Science and Systems*, volume 2, pages 2–1.

Frank E Ritter, Farnaz Tehranchi, and Jacob D Oury. 2019. ACT-R: A cognitive architecture for modeling cognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, 10(3):e1488.

Hugo Romat, Mary-Anne Williams, Xun Wang, Benjamin Johnston, and Henry Bard. 2016. Natural human-robot interaction using social cues. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*.

Matthias Scheutz, Thomas Williams, Evan Krause, Bradley Oosterveld, Vasanth Sarathy, and Tyler Frasca. 2019. An overview of the distributed integrated cognition affect and reflection DIARC architecture. In *Cognitive Architectures*.

Sarath Sreedharan, Siddharth Srivastava, and Subbarao Kambhampati. 2018. Hierarchical expertise level modeling for user specific contrastive explanations. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*.

Cristen Torrey, Aaron Powers, Matthew Marge, Susan R Fussell, and Sara Kiesler. 2006. Effects of adaptive robot dialogue on information exchange and social relations. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*.

Ning Wang, David V. Pynadath, and Susan G. Hill. 2016. Trust calibration within a human-robot team: Comparing automatically generated explanations. In *11th ACM/IEEE International Conference on Human-Robot Interaction*.