Human, we have a problem: What to say when things go wrong

Mihai Pomarlan
Vanja Sophie Cangalovic
Robert Porzel
John Bateman*
pomarlan@uni-bremen.de
vanja@uni-bremen.de
porzel@uni-bremen.de
bateman@uni-bremen.de
University of Bremen

1 MOTIVATION

Failure is a fact of life. Doubly so for robots it seems, as anyone who has spent sleepless nights preparing a demonstration for a review meeting will know. The spatula that is right in front of the robot's nose is recognized as an orange. The lid that one must use, lest hot popcorn fly out all over, is dropped on the floor before it reaches its destination. After three minutes, motion planning finally gives up and announces there is no path to place a cup on an otherwise completely empty table.

Such cartoonish-sounding errors will hopefully be worked out of a robotic system if it is to have any hope of deployment in a semi-structured, consumer environment such as a home or nursing ward. Nevertheless, one can expect that errors will occasionally happen– and also that a human being, watching the robot perform, would often not be able to tell what exactly went wrong just by looking at the robot. Further, sometimes the robot might need to ask for help, which in turn opens up issues of whether it is able, concisely, to explain to a human what its request is.

Obviously, minimizing error and the necessity to call in human assistance is a hot topic in many areas of robotics research. Our not too controversial hypothesis is that this minimization will never drive failures to zero, and that some failures will prove beyond the robot's capabilities to recover from.

Because of this, we propose an NLG module to report on failures and, if necessary, ask for assistance, which is to be constructed around an ontology of failures [4] and recovery strategies (in progress, yet to be published work). Failure handling is a normal part of robot behavior and the mentioned ontology addresses competency questions related to classifying failures and selecting recovery strategies for them as part of a normal robotic perception-action loop.

Until this point we have relied on the reader's intuition for what failure means, and given some examples. To proceed, we will first define "failure" as *a label that can be applied to an Event, which communicates that there is an Agent who intended the Event to instantiate a Task with a particular Goal, which is not met by the Event's Outcome*. Both Goal and Outcome are Descriptions of the state of the world at some level of granularity.

2 COMMUNICATIVE GOALS IN CASE OF TASK FAILURE

Competency questions are used in ontology engineering to describe what sort of reasoning queries an ontology can support. Similarly, we will present here a tentative list of communicative goals that an NLG system should enable when describing failures to a human. In some cases, the analogy with ontology competency questions will be very direct, as the communicative goal will correspond to a competency question of the underlying failure ontology. We will in each case give a short description of how ongoing research in our collective will feed towards formalizing procedures to obtain the knowledge an NLG system would need to achieve these communicative goals.

2.1 Does the robot realize there is a failure at all?

Perception of a failure event may itself fail, e.g. when a robot keeps moving to a destination without realizing that the item it is supposed to carry has been dropped. At the most basic, a robot's reporting on a failure should indicate to a listener that the robot is aware something bad has happened, so that both agents understand they are in some sort of recovery situation. That is, the robot knows of the failure, the listener knows the robot knows, and potentially shared control over the situation can be established if the robot doesn't figure a way out; see the subsection on delegation, 2.5, below.

2.2 Can the robot categorize the failure?

As suggested by the definition, a failure is describable in terms of what task was unsuccessfully attempted and/or in terms of how the outcome is incompatible with the goal. The failure ontology of [4] supports this by offering an axiomatization of failure types. By the time the NLG component would be involved, classification reasoning would already have been carried out, if necessary– usually, various components produce error signals, and all the reasoner needs to do is map the error signal to the corresponding failure class.

2.3 Can the robot explain why the failure happened?

It has been observed that human beings prefer explanations in terms of causal narratives [3, 7, 8], so explanations should follow such a form. We think this in particular is an important direction to pursue in robotics NLG, as it is a key to obtaining useful explanations that

communicate a trustworthy, human-understandable process. Obtaining and communicating causal understanding of situations is a major goal that we pursue in our work related to service robots.

The failure ontology already distinguishes several facets that a failure may have, such as a rough classification of the physical mechanisms involved (e.g. electrical, mechanical, communication channel etc.), the temporal placement of a failure along the execution of a task (pre-, during, post-), whether the failure is related to the presence or absence of a resource or capability and so on. This information about a failure can be converted into material or effective causes to explain why the failure was observed.

Further, reasoning about failure cause is already in development in the failure and recovery ontology for an independent problem: selecting a recovery method. Some failures can be undone by addressing their consequences– e.g a dropped item can be picked up–, but for others an underlying causal mechanism must be stopped– e.g. a device that refuses to turn on must be plugged in first.

2.4 Can the robot select a recovery method and what does it plan to do?

As mentioned, in our ongoing work we already distinguish between failures created by a sustained cause which must be removed to address the failure vs. failures where the consequences of an action can simply be undone. There is also a third method, "delegation", which is to be used only when the other methods are deemed infeasible. In case of an error-cause-elimination or consequence-undoing method, the details of the procedure may or may not be interesting to a human observer, but it is important to communicate whether the robot has some idea of what to do by announcing it has a plan. Optionally, it may be useful to articulate what the plan's *instrumental goal* is, that is, a top-level goal which if achieved will correct the failure or prevent the failure cause from persisting.

2.5 Assuming delegation is chosen as a recovery method, can the robot explain what it wants the human to do, and why?

Recovery methods produce planning queries for the robot's own planning and/or libraries of stored programs, so a formal description of an instrumental goal would be available; see section 2.4.

Explanation of the delegation-why meanwhile can fall back to somewhat stereotypical patterns. Typically, delegation is needed when the robot lacks some set of capabilities or some relevant resources; knowledge itself may be a resource. Therefore, pinpointing the missing resource or capability is often a sufficient explanation.

3 APPROACH

Ontologies are explicit conceptualizations used to define a shared language between several software modules, hence at the core of our approach we have an ontology; in this case the failure and recovery ontology [4], together with an ontology of linguistic structures called the Generalized Upper Model [1, 6], under the foundational framework of DOLCE UltraLite [9, 10]. This architecture leads to bidirectional/symmetric linguistic capabilities using the same knowledge engineering: everything that is producible, must be parseable and vice versa; e.g. the human feedback must be understood.

A knowledge-driven approach allows maintaining a robot belief state that is easily transferable and repurposable between modules. The belief state may contain sub-symbolic data such as trajectory or sensor data, but here we will focus on symbolic entities such as tasks and situations. Situations are, in DUL terminology, an interpretation which an agent can use to describe the relations between entities in the world. Objects may play various roles in a situation, which makes them analogous to semantic frames [5].

Producing a text is triggered in particular situations; one such situation is the occurrence of a failure. In this case, the knowledge graph of what the robot believes the current situations are serves as input to an NLG pipeline. This graph includes results coming from reasoning queries about failure cause, recovery strategy, and status of planning tasks triggered by recovery methods as part of the robot's usual perception-action loop.

A content selection heuristic then selects items from this graph to generate messages. Several kinds of messages may be considered, such as announcing a failure, explaining a failure, announcing a recovery method, explaining a delegation goal. Items in the knowledge graph must be tagged as to which purposes they are relevant for; this is also achievable by DL classification and subsumption reasoning.

At this time, we have yet to experiment with content selection heuristics. These however can, and should make use of taxonomic information in the ontology, since this helps formalize the level of abstraction at which a message is generated.

Finally, a KPML [2] logical form is constructed based on the selected content, and then KPML produces a text realization.

There are of course other pragmatic considerations, chief among them the speed of reasoning. So far, our competency questions appear possible to reformulate in small, efficiently decidable fragments of DL such as EL and ELI, which should translate in the DL reasoning itself not being a bottleneck.

REFERENCES

[1] J. Bateman, J. Hois, R. Ross, T. Tenbrink, and S. Farrar. 2008. The Generalized Upper Model 3.0: Documentation. SFB/TR8 internal report, Collaborative Research Center for Spatial Cognition, University of Bremen, Germany.

[2] John Bateman. 1997. *Enabling technology for multilingual natural language generation: the KPML development environment*, volume 3(1):15–55. Journal of Natural Language Engineering.

[3] Or Biran and Kathleen McKeown. 2017. Human-centric justification of machine learning predictions. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1461–1467.

[4] Mohammed Diab, Mihai Pomarlan, Daniel Beßler, Aliakbar Abkari, Jan Rossel, John Bateman, and Michael Beetz. 2019. An ontology for failure interpretation in automated planning and execution. In *Fourth Iberian Robotics Conference*, ROBOT '19, Porto, Portugal.

[5] C.J. Fillmore. 1982. Frame Semantics. In Linguistics Society of Korea, editor, *Linguistics in the morning calm*, pages 111–137. Hanshin, Seoul.

[6] J. Hois, Thora Tenbrink, R. Ross, and J. Bateman. 2009. GUM-Space – The Generalized Upper Model spatial extension: a linguistically-motivated ontology for the semantics of spatial language. SFB/TR8 technical report, Collaborative Research Center for Spatial Cognition, University of Bremen, Germany. Version 3.0.

[7] Daniel Kahneman. 2011. *Thinking, fast and slow*. Farrar, Straus and Giroux, New York.

[8] Frank C. Keil. 2006. Explanation and understanding. *Annual Review of Psychology*, 57(1):227–254.

[9] Viviana Mascardi, Valentina Cordì, and Paolo Rosso. 2010. Technical report disi-tr-06-21, university of genua.

[10] C Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, and A Oltramari. 2003. Wonderweb deliverable d18 ontology library.