PentoRob: A Puzzle-Playing Robot for Dialogue Experiments

Julian Hough and David Schlangen
Dialogue Systems Group // CITEC // Faculty of Linguistics and Literature
Bielefeld University
firstname.lastname@uni-bielefeld.de

Abstract

We present a simple puzzle-playing interactive robot, PentoRob, which allows investigation into real-time, real-world dialogue. The dialogue control framework consists of a combination of interactive Harel statecharts and the Incremental Unit framework. We outline its architecture and potential use cases for dialogue and human-robot interaction.

1 Introduction

In embodied dialogue systems research, there is a need for simple robots that do not require heavy mechanical maintenance or robotics experts when developing functionality of interest. Here we present a system to fulfil these needs: *PentoRob*, a simple pick-and-place robot controlled by an incremental dialogue framework.

2 PentoRob

PentoRob is a puzzle-playing robot which manipulates Pentomino pieces– see Fig. 1. Its dialogue control consists of Harel statecharts (Harel, 1987) and the Incremental Unit framework (Schlangen and Skantze, 2011), and is implemented with the dialogue toolkit InproTK (Baumann and Schlangen, 2012). Here we describe its components in terms of input information or Incremental Units (IUs), processing, and output IUs.

Hardware For the robotic arm, we use the ShapeOko2,[1] a heavy-duty 3-axis CNC machine, which we modified with a rotatable electromagnet, whereby its movement and magnetic field is controlled via two Arduino boards. The sensors are a webcam and microphone.

Incremental Speech Recognizer (ASR) We use Google's web-based ASR API which packages hypotheses into individual *WordIUs*. While

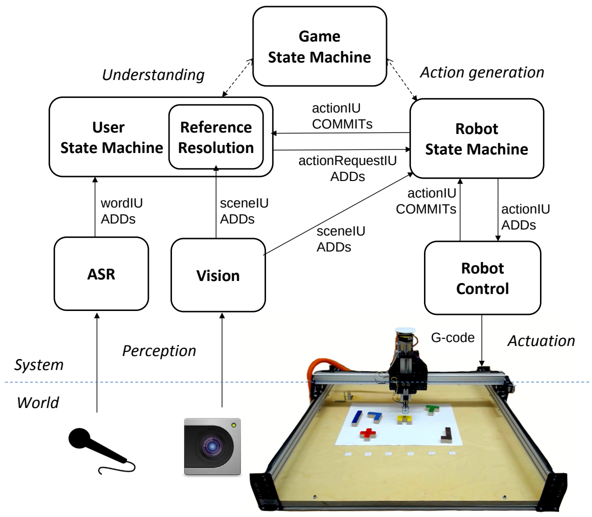

Figure 1: PentoRob's architecture.

its incremental performance is not as responsive as more inherently incremental local systems such as Kaldi or Sphinx-4, this does not incur great costs for many interesting applications.

Computer Vision (CV) We use OpenCV in a Python module to track objects in the camera's view. This information is relayed to InproTK from Python via the Robotics Service Bus (RSB),[2] which outputs IDs and positions of objects detected along with their low-level features (e.g., RGB/HSV values, x,y coordinates, number of edges, etc.), converting these into $SceneIU$s which the reference resolution module consumes and the *Robot State Machine* uses for obtaining the positions of objects it plans to grab.

Reference resolution (WAC) The reference resolution component consists of a Words As Classifiers (WAC) model (Kennington and Schlangen, 2015) trained on real-world objects using low-level vision features from $SceneIU$s.

[1] `http://www.shapeoko.com.`

[2] `https://code.cor-lab.de/projects/rsb.`

Appears in *JerSem: The 20th Workshop on the Semantics and Pragmatics of Dialogue,*
edited by Julie Hunter, Mandy Simons and Matthew Stone. New Brunswick, NJ USA,
July 16-18, 2016, pages 114-115.

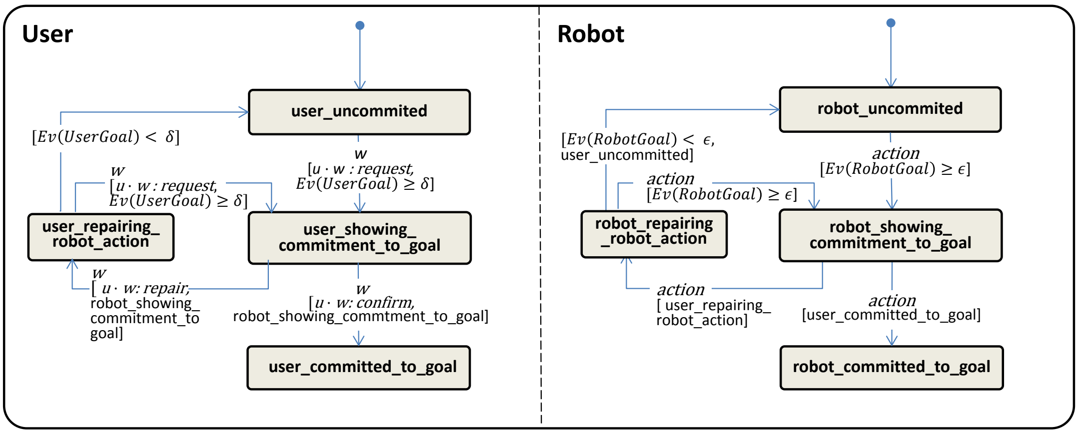

Figure 2: PentoRob's Interactive Statechart with two parallel, concurrent states

During application, as a referring expression is recognised, each classifier for the words in the expression are applied to the puzzle pieces in view, which after normalisation, results in a probability distribution over pieces.

User and Robot State Machines For dialogue control, we use an *Interactive Statechart*– see Fig. 2. Rather than comprising a single dialogue state, there are concurrent states for each agent in the interaction with their own variables. The *User State Machine* has access to the estimated current user goal $UserGoal$ and a *strength-of-evidence* function $Ev(UserGoal)$, both of which can be defined by the designer. In our domain $UserGoal$ is the taking of the most likely object according to WAC's output distribution given the utterance u so far and the Ev function as the probability value of the highest ranked object in WAC's distribution over its second highest rank. If $UserGoal$ is changed or instantiated, a new *ActionRequestIU* is made available in its right buffer with the goal.

The *Robot State Machine* gets access to its transition conditions involving the user through the ActionRequestIUs. Through a simple planning function, a number of ActionIUs are cued to achieve the goal. It sends these as RSB messages to the PentoRob control module and once confirmed, via RSB, that the action has begun, the ActionIU is *committed*. For estimation of its own state, the robot state has a strength-of-evidence function $Ev(RobotGoal)$ defined by the designer.

PentoRob control module The module controlling the robotic actuation of the ShapeOKO arm is a Python module with an Arduino board G-code interface. This sends RSB feedback messages to the Robot State Machine to the effect that actions have been successful or unsuccessful.

3 Use cases

We are currently experimenting with achieving more fluidity for grounding behaviour in human-robot interaction. The statechart in Fig. 2 has parameters δ and ϵ which are thresholds for the $Robot$ and $User$ that must be reached by the Ev functions to show sufficient evidence of each agent's goal. Early results show that lower thresholds allow more fluid grounding behaviour, while higher thresholds are 'safer' for task success. We are planning a series of related experiments. Other areas where our setup could be used is learning grounded semantics for verbs.

Acknowledgments

We thank Oliver Eickmeyer, Casey Kennington and Livia Dia for contributions to software and hardware. This work was supported by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University, funded by the German Research Foundation (DFG), and the DFG-funded DUEL project (grant SCHL 845/5-1).

References

Timo Baumann and David Schlangen. 2012. The inprotk 2012 release. In *NAACL-HLT Workshop on Future Directions and Needs in the Spoken Dialog Community: Tools and Data*. ACL.

David Harel. 1987. Statecharts: A visual formalism for complex systems. *Science of computer programming*, 8(3).

Casey Kennington and David Schlangen. 2015. Simple learning and compositional application of perceptually grounded word meanings for incremental reference resolution. Proceedings of the Conference for the Association for Computational Linguistics (ACL). ACL.

David Schlangen and Gabriel Skantze. 2011. A General, Abstract Model of Incremental Dialogue Processing. *Dialoge & Discourse*, 2(1).