Investigating Fluidity for Human-Robot Interaction with Real-time, Real-world Grounding Strategies

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 real-time, real-world grounding framework, and a system which implements it in a simple robot, allowing investigation into different grounding strategies. We put particular focus on the grounding effects of non-linguistic task-related actions. We experiment with a trade-off between the fluidity of the grounding mechanism with the 'safety' of ensuring task success. The framework consists of a combination of interactive Harel statecharts and the Incremental Unit framework. We evaluate its in-robot implementation in a study with human users and find that in simple grounding situations, a model allowing greater fluidity is perceived to have better understanding of the user's speech.

1 Introduction

Developing suitable grounding mechanisms for communication in the sense of (Clark and Brennan, 1991; Clark, 1996) is an ongoing challenge for designers of robotic systems which interpret speech. If grounding is the way in which interaction participants build and align their internal representations towards shared information or 'common ground', given the vastly different internal representations of humans and robots, one might concede the title of Kruijff (2012)'s paper: 'There is no common ground in human-robot interaction'.

However despite the lack of 'real' common ground, a robot can still understand what the user means 'to a criterion sufficient for current purposes' (Clark and Brennan, 1991) at a given point in the interaction, if it is equipped with grounding mechanisms which deal with the inherent uncertainty in situated dialogue for a robot. This uncertainty lies at multiple layers, including the recognition of words, object recognition and tracking, resolving references to the objects, the recognition of the user's intentions, and the success in execution of robotic actions. Furthermore, if we are to reach beyond task completion or speed as criteria for interactive success and wish the interaction to be more 'fluid', these grounding mechanisms must operate continuously in real time as robotic actions or user utterances are in progress.

In this paper, we present a simple real-time, real-world grounding framework, and a system which implements it in a simple robot, allowing investigation into different grounding strategies. Here, we experiment with a trade-off between the fluidity of the grounding mechanism with the 'safety' of ensuring task success. The framework consists of a combination of interactive Harel statecharts (Harel, 1987) and the Incremental Unit framework (Schlangen and Skantze, 2011), and is implemented in dialogue toolkit InproTK (Baumann and Schlangen, 2012).

2 Achieving Fluid Communicative Grounding in Dialogic Robots

In this paper we are concerned with a simple pick-and-place robot with uni-modal communication abilities, which is simply its manipulation behaviour of objects– see Fig. 1 for example utterances from user U and system S's actions. While our robot does not have natural language generation (NLG) capabilities, its physical actions are first class citizens of the dialogue so it is capable of dialogic behaviour through action.

As mentioned above, while a human and robot's internal representations of a situation can differ inherently, success is possible through recovery

A. Non-incremental grounding:

(1) U: Put the red cross in box 2 right
S: [moves to x] [grabs x] [moves to box 2] [drops x]

(2) i) U: Put the red cross in box 2 no, the other red cross
S: [moves to x] [grabs x] [moves to box 2]

ii) U: right
S:[moves to x's original position][drops x][moves to y][grabs y][moves to box 2] [drops y]

B. Incremental grounding:

(3) U: Take the red cross right put it in box 2 right
S: [moves to x] [grabs x] [moves to box 2] [drops x]

(4) U: Take the red cross no the other one right put it in box 2 right
S: [moves to x] [moves to y] [grabs y] [moves to box 2] [drops y]

C. Fluid incremental grounding, allowing concurrent user speech and robotic action:

(5) U: Take the red cross right put it in box 2 right
S: [moves to x][grabs x] [moves to box 2] [drops x]

(6) U: Take the red cross no the other one right put it in box 2 right
S: [moves to x(aborted)][moves to y][grabs y] [moves to box 2] [drops y]

Figure 1: Grounding modes in a robotic dialogue system that manipulates real-world objects.

from misunderstanding, which has been central to dialogue systems research (Traum, 1994; Traum and Larsson, 2003), with recent work showing how this can operate incrementally (see e.g. (Buß and Schlangen, 2011; Skantze and Hjalmarsson, 2010)), and in situated dialogue domains, through simulation with virtual agents (Marge and Rudnicky, 2011; Raux and Nakano, 2010; Buschmeier and Kopp, 2012). In robotics, much of the grounding research has focussed on perspective taking and frame of reference differing between robot and human (Liu et al., 2010; Liu et al., 2012; Kollar et al., 2010).

The aspect of grounding we focus on here is the mechanisms needed for it to be done fluidly in real time. In line with results from human-human interaction where action is shown to be representative of the current state of understanding with little latency (Tanenhaus and Brown-Schmidt, 2008; McKinstry et al., 2008) and where moving in response to instructions happens *before* the end of the utterance (Hough et al., 2015), we hypothesized that the greater the fluidity, the more natural the robot's action would appear. To illustrate, in Fig. 1, we show three modes of grounding, (A) non-incremental, (B) incremental and (C) fluid. Each mode has the ability to recognize positive feedback and repair and deal with it appropriately, however (A) only allows grounding in a 'half-duplex' fashion with no overlapping speech and robot action, and grounding can only be done once a completed semantic frame for the current user's intention has been interpreted. When the entire frame has been recognized correctly, the user waits until the robot has shown complete understanding of the user's intention through moving to the target area and awaits confirmation to drop the object there. In recovering from misunderstanding as in (2) when the user repairs the robot's action, not only must the current action be 'undone' but the new action must then also be carried out from the beginning, resulting in long periods of waiting for the user. In mode (B), grounding again happens in a half-duplex fashion, however with opportunities for grounding after shorter increments of speech and with partial information about the user's overall goal– the benefit for repair and recovery incrementally is clear in (4). In (C), the grounding again happens incrementally, however in a full-duplex way, where concurrency of speech and action is allowed and reasoned with appropriately.

To allow human-robot interaction to be more like mode (B) rather than (A), appropriate mechanisms can be designed for robots in line with computational theories of grounding (Traum, 1994; Traum and Larsson, 2003; Ginzburg, 2012), adjusting these mechanisms to work in real time rather than turn-finally, in line with recent work on incremental grounding theories (Ginzburg et

al., 2014; Eshghi et al., 2015) where semantic frames can be grounded partially as an utterance progresses. To move towards fluid mode (C), this type of incremental processing not only requires incremental interpretation word-by-word, but use of the context at the exact time each word is recognized, where here, context consists in the estimation of both the user's state and the robot's current state through self-monitoring, both of which can change dynamically during the course of an utterance, or even during a word. In this setting, during a repair from the user, the robot must reason about the action currently 'under discussion' and abort it as efficiently as possible in order to switch to an action consistent with the new goal presented by the user. This self-repair of action involves an estimation of which part of the action the user is trying to repair. The same is true of the converse of repair, where positive confirmations like 'right' may need to be interpreted before the robot has shown unambiguously what its goal is to allow the fluidity in setting (C)– this requires a self-monitoring process which estimates at which point the robot has shown its goal *sufficiently* clearly to the user, during its movement and not necessarily only after its goal has become completely unambiguous.

3 Interactive Statecharts and the Incremental Unit Framework for Real-time Grounding

Our approach to modelling and implementing real-time grounding mechanisms follows work using Harel statecharts (Harel, 1987) for dialogue control in robotic dialogue systems by (Peltason and Wrede, 2010; Skantze and Al Moubayed, 2012). However here, rather than characterizing a single dialogue state which is accessed by a single dialogue manager, our statechart characterizes two independent parallel states for the user and robot, taking an agents-based approach in the sense of (Jennings, 2001).

As illustrated in the diagrams in Fig. 2 and Fig. 7 (Appendix), as per standard statecharts we utilize *states* (boxes) and *transitions* (directed edges) which are executable by *trigger events* (main edge labels) and *conditions* (edge labels within []), and, additionally triggered *actions* can be represented either within the states (the variable assignments and DO statements in the body of the boxes), or on the transition edges, after /. We dub these *Interactive Statecharts* as the transitions in the participant states can have triggering events and conditions referring to the other interaction partner's state.

We also make use of *composite* states (or superstates) which generalize two or more substates, shown diagrammatically by a surrounding box, which modularizes, reducing the need to define the transitions for all substates, and diagrammatically reduces the number of arrows.

We also refer to variables for each agent state, which for our purposes are *UserGoal* and *RobotGoal*– these represent each agent's current private goal as estimated by the robot (i.e. this is not an omniscient world view).

Given there are mutual dependencies between the two parallel states, one could argue the statechart obscures the complexity which a Finite State Machine (FSM) characterization of the dialogue state would make explicit, and without converting them to FSMs, estimating the probability distributions for the whole composite state is less straight-forward. However, the extra expressive power makes modelling interactive situations and designing grounding mechanisms much simpler. We discuss how to deal with concurrency problems in §3.2, and discuss probabilistic state estimation in the final discussion, though it is not the main focus of this paper.

3.1 A simple concurrent grounding model

To provide a grounding mechanism for robots to achieve more fluid interaction, we characterize the user and robot as having *parallel* states (either side of the dotted line) – see Fig. 2. This allows modelling the concurrent robot and human states the robot believes they are in during the interaction without having to explicitly represent the Cartesian product of all possible dialogue states.

Fig. 2 defines the grounding states and transitions for a simple robotic dialogue system which interprets a user's speech to carry out actions. The main motivation of the model is to explore the nature of the criteria by which the robot judges both their own and their interaction partner's goals to have become publicly manifest (though not necessarily grounded) in real time, and therefore when they are *showing commitment* to them. To evaluate whether the criteria have been met we posit functions Ev for each agent's state, which is a strength-of-evidence valuation that the agent has

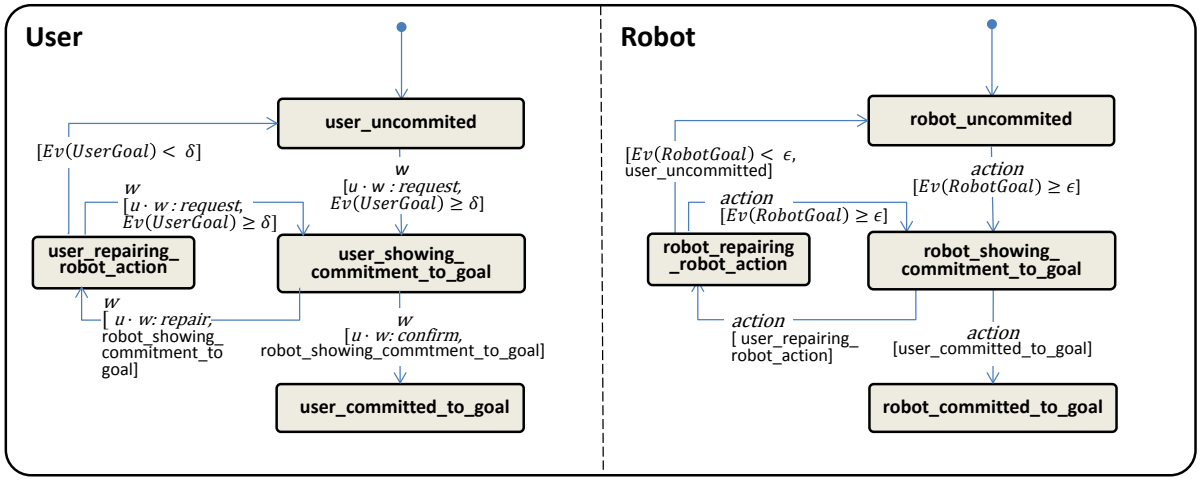

Figure 2: An Interactive Statechart as modelled by the Robot. The statechart consists of two parallel, concurrent states, one for each participant. The triggering events and conditions in the transition functions (the directed edges) can reference the other state.

displayed their goal publicly, where goals are hidden in the case of the user state and observed in the case of the robot.

As shown in Fig.7, $UserGoal$ is estimated as the most likely desired future state the user intends in the set of possible future states $States$, given the current utterance u, the robot's state $Robot$ and the current task's state $Task$, as below.

$$UserGoal := \underset{s \in States}{\arg\max}\ p(s \mid u, Robot, Task) \tag{7}$$

Note, conditioning on the current task is in line with agenda-based approaches to dialogue management (Traum and Larsson, 2003) and also in line with characterizing tasks (or games) as state machines themselves. Our future work will involve more complex task structures.

While the user's goal is being updated through new evidence, this goal can only be judged to become sufficiently mutually manifest with the robot when a certain confidence criteria has been met– here we characterize this as reaching a real-valued threshold δ. As the statechart diagram shows, once $Ev(UserGoal) \geq \delta$ then the state `user_showing_commitment_to_goal` substate can be entered, which is accessible by the Robot state machine in its transition functions to trigger the robot into `robot_showing_commitment_to_goal`. Characterizing this criteria as a threshold allows experimentation into increasing responsiveness of the robot by reducing it, and we explore this in our implemented system– see §5 below.

Conversely, the Robot's view of its own state uses the function $Ev(RobotGoal)$ and its own threshold ϵ. Unlike the user, the robot's own state is taken to be fully observed, however it must still estimate when its own $RobotGoal$ is made public by its action, and once ϵ has been reached, the robot may enter `robot_showing_commitment_to_goal`. Once this is the case it is permissible for the user state to either commit to the goal and trigger grounding, else engage the robot in repair. The robot will be in the repairing state until the user's state has exited the `user_repairing_robot_action` state. Note that it is only possible for the user state to repair the $RobotGoal$, rather than $UserGoal$– the user can repair the latter through self-repair, but that is currently not represented as its own state.

The necessary conditions of incrementality posed by examples in Fig. 1 (B) and (C) above are met here as the increment size of the triggering events in the $User$ state is the utterance of the latest word w in current utterance u (as opposed to the latest complete utterance). The principal Natural Language Understanding (NLU) decisions are therefore to classify incrementally which type of dialogue act u is, (e.g. $u : Confirm$), whether w begins a new dialogue act or not, and estimate $UserGoal$. The statechart is then checked to see if a transition is possible from the user's current state as each word is processed, akin to incremental dialogue state tracking (Williams, 2012).

3.2 Managing Fluid Grounding with the IU framework

To manage the processing and information flow, we use the Incremental Unit (IU) framework (Schlangen and Skantze, 2011). Currently, in implemented IU framework systems such as Jindigo (Skantze and Hjalmarsson, 2010), DyLan (Purver et al., 2011) and InproTK (Baumann and Schlangen, 2012), processing goes bottom-up (from sensors to actuators) and the creation of incremental units (IUs) is driven by input events to each module from bottom to top. IUs are packages of information at a pre-defined level of granularity, for instance a *wordIU* can be used to represent a single incremental ASR word hypothesis, and their creation in the output buffers of a module triggers downstream processing and creation of new IUs in modules with access to that buffer. IUs can be defined to be connected by directed edges, called *Grounded In* links, which in general take the semantics of "triggered by" from the source to the sink.

Grounded In links are useful in cases where input IU hypotheses may be *revoked* (for instance, by changing ASR hypotheses), as reasoning can be triggered about how to revoke or repair actions that are Grounded In these input IUs. Buß and Schlangen (2011) take precisely this approach with their dialogue manager DIUM, and Kennington et al. (2014) show how abandoning synthesis plans can be done gracefully at short notice.

In order to manage the grounding strategies above, we recast the IU dependencies: while the output IUs are taken as Grounded In the input IUs which triggered them, as per standard processing, in our system the reverse will also be true: consistent with the statecharts driving the behaviour, the interpretation of a user action is taken as an action in response to the robot's latest or currently ongoing robot action, consequently interpretation IUs can be grounded in action IUs– see the reversed feedback arrow in Fig. 3.

To deal with concurrency issues that this closed-loop approach has, the IU modules coordinate their behaviours by sending event instances to each other, where events here are in fact IU edit messages shared in their buffers. The edit messages consist in *ADD*s where the IU is initially created, *COMMIT*s if there is certainty they will not change their payload, and, as mentioned above *REVOKE*s may be sent if the basis for an ADDed

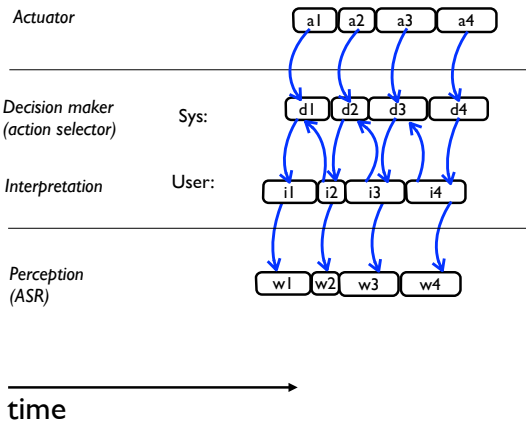

Figure 3: The addition of tight feedback over standard IU approaches helps achieve requirements of fluid interaction and situated repair interpretation. Grounded In links in blue.

IU becomes unreliable. IUs also have different temporal statuses of being either *upcoming*, *ongoing* or *completed*, a temporal logic which allows the system to reason with the status of the actions being executed or planned by the robot.

4 PentoRob: A Simple Robot for Investigating Grounding

We implement the above grounding model and incremental processing in a real-world pick-and-place robot *PentoRob*, the architecture of which can be seen in Fig. 4. The domain we use in this paper is grabbing and placing real-world Pentomino pieces at target locations, however the system is adaptable to novel objects and tasks.

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.

4.1 System components

PentoRob was implemented in Java using the InproTK (Baumann and Schlangen, 2012) dialogue systems toolkit.[2] The modules involved are de-

[1] http://www.shapeoko.com/wiki/index.php/ShapeOko_2

[2] http://bitbucket.org/inpro/inprotk

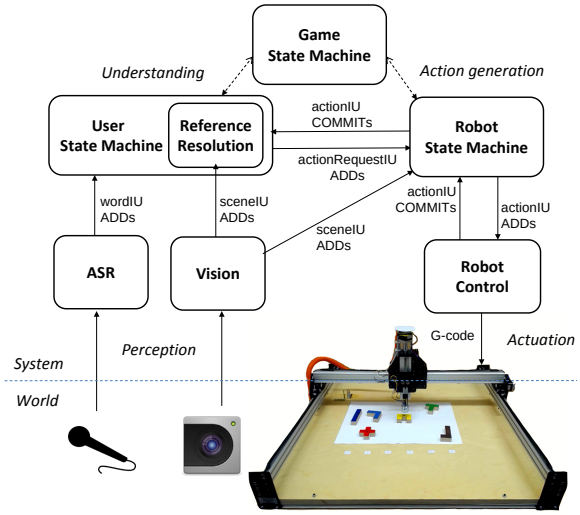

Figure 4: PentoRob's architecture.

scribed below, in terms of their input information or IUs, processing, and output IUs.

Incremental Speech Recognizer (ASR) We use Google's web-based ASR API (Schalkwyk et al., 2010) in German mode, in line with the native language of our evaluation participants. As Baumann et al. (2016) showed, while Google can produce partial results of either multiple or single words, all outputs are packaged into single *WordIUs*. Its incremental performance is not as responsive as more inherently incremental local systems such as Kaldi or Sphinx-4, however, even when trained on in-domain data, other systems cannot consistently match its Word Error Rate in our target domain in German, where it achieves 20%. Its slightly sub-optimal incremental performance did not incur great costs in terms of the grounding we focus on here.

Computer Vision (CV) We utilize 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),[3] which outputs IDs and positions of objects it detects in the scene 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 downstream reference resolution model consumes.The Robot State Machine also uses these for reasoning about positions of the objects it plans to grab.[4]

Reference resolution (WAC) The reference resolution component consists of a Words As Classifiers (WAC) model (Kennington and Schlangen, 2015). PentoRob's WAC model is trained on a corpus of Wizard-of-Oz Pentomino puzzle playing dialogue interactions. In off-line training, WAC learns a functional "fit" between words in the user's speech and low-level visual object features, learning a logistic regression classifier for each word. Once trained, when given the context of a novel visual scene and novel incoming words, each word classifier yields a probability given each object's features. During application, as a referring expression is uttered and recognised, each classifier for the words in the expression are applied to all objects in the scene, which after normalisation, results in a probability distribution over objects. Kennington and Schlangen (2015) report 65% accuracy on a 1-out-of-32 reference resolution task in this domain with the same features. For this paper, this accuracy can be seen as a lower bound, as the experimental set-up we report below uses a maximum of 6 objects, where the performance is generally significantly better.

User State Machine We implement the principal NLU features within the User State Machine module, which constitutes the $User$ state of the Interactive Statechart. While the statechart manages the possible transitions between states, their triggering criteria require the variables of $UserGoal$, the estimated current user goal and its strength-of-evidence function Ev to be defined. In our domain we characterize $UserGoal$ as simply taking or placing most likely object in the referent set R being referred to according to WAC's output distribution given the utterance u so far, e.g. (8), and the Ev function as simply the probability value of the highest ranked object in WAC's distribution over its second highest rank as in (9).

$$UserGoal = TAKE(\arg\max_{r \in R} p(r \mid u)) \quad (8)$$

$$Ev(UserGoal) = \text{Margin}(\arg\max_{r \in R} p(r \mid u)) \quad (9)$$

As for the process which feeds incoming words into the WAC model to obtain $UserGoal$, here

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

[4] The objects' positions are calculated accurately from a single video stream using perspective projection.

we use a simple incremental NLU method which is sensitive to the $Robot$'s current state in addition to the $User$ statechart. This is a process which first performs sub-utterance dialogue act (DA) classification, judging the utterance to be in $\{request, confirm, repair\}$ after every word. The classifier is a simple segmenter which uses key word spotting for $confirm$ words and common $repair$ initiating words, and also classifies a $repair$ if the word indicates change in the $UserGoal$ as defined in (8), else outputting the default $request$.[5] Given the DA classification, the state machine is queried to see if transitioning away from the current state is possible according to the statechart (see Fig. 7 in the Appendix)– if not it remains in the same state and treats the user's speech as irrelevant.

If a successful state change is achieved, then if $UserGoal$ has changed or been instantiated in the process, a new *ActionRequestIU* is made available in its right buffer, whose payload is a frame with the dialogue act type, the action type (`take` or `place`) and optional arguments `target_piece` and `target_location`.

For dealing with repairs, as seen in Fig. 7, entering a repairing state triggers a prune of $States$, removing the evidenced $RobotGoal$. In PentoRob this is simply a pruning of the referent set R of the objects(s) in the $RobotGoal$ as below:

$$R := \{x \mid p(RobotGoal \mid x) = 0\} \qquad (10)$$

This simple strategy allows $UserGoal$ to be recalculated, resulting in interactions like (4) and (6) in Fig.1.

Robot State Machine The Robot's state machine gets access to its transition conditions involving the User's state machine through the ActionRequestIUs it has access to in its left buffer. As seen in Fig.7 (Appendix), when the $User$ state is `showing_commitment_to_goal`, the $RobotGoal$ is set to $UserGoal$, and through a simple planning function, a number of ActionIUs are cued to achieve it – it sends these as RSB messages to the PentoRob actuation module and once confirmed, again via RSB, that the action has begun, the ActionIU is *committed* and the Robot's action state is set to one of the following, with superstates in brackets:

[5]While a somewhat crude approach, it worked reliably enough in our test domain, and is not the focus of the paper.

```
{stationary_without_piece |
moving_without_piece |
moving_to_piece (taking) |
over_target_piece (taking) |
grabbing_piece (taking) |
stationary_with_piece(placing) |
moving_with_piece (placing) |
over_target_location (placing) |
dropping_piece (placing)}
```

For estimation of its own state, the robot state has the following function:

$$Ev(RobotGoal) = \begin{cases} 1 & \text{if over\_target\_piece,} \\ 1 & \text{if over\_target\_location,} \\ 0.5 & \text{if taking,} \\ 0.5 & \text{if placing,} \\ 0 & \text{otherwise} \end{cases} \qquad (11)$$

The simplistic function embodies the assumption that there is absolute certainty that PentoRob's goal has been demonstrated when its arm is directly over the target pieces and locations, else if it is moving to these positions, there is some evidence, else there is none.

PentoRob actuation module The module controlling the actual robotic actuation of the ShapeOKO arm is a Python module with an Arduino board G-code interface to the arm. This sends RSB feedback messages to the PentoRob control module to the effect that actions have been successful or unsuccessfully started, and with their estimated finishing time.

5 Evaluation Experiments

With the above system, we can successfully achieve all three types of grounding strategy in Fig 1. We evaluate the incremental mode (B) and fluid mode (C) in a user study with German speakers. In our first and principal study we experiment with varying the $Robot$ state's ϵ grounding parameter to see whether users show preference for a more fluid model, and what effect fluidity has on task success.

The study was a within-subjects design. It had 12 participants, who played a total of 6 rounds each of a simple game with PentoRob. Users were instructed to tell the robot to pick up and place wooden Pentomino pieces onto numbered locations at the bottom of the playing board in a given order according to a photograph of final configurations showing the final location and the desired order of placement. Participants were told they could confirm or correct Pentorob's actions.

They played three rounds in progressing level of difficulty, beginning with a simple situation of 3 pieces of all differing shapes and colours arranged in a line and far apart, followed by another round with 4 pieces arranged in a non grid-like fashion, followed by a more difficult round with 6 pieces where the final two shapes to be placed were close together and the same colour. They play each round twice, once with each version of the system. The order of the conditions was changed each time. The two settings PentoRob's system operated in were as follows:

> **Incremental:** A cautious strategy whereby ϵ = 1. Given (11) only allows PentoRob to enter the `robot_showing_commitment_to_goal` state when in the states `over_target_piece` or `over_target_location`, confirmations and repairs cannot be interpreted during robotic action.
>
> **Fluid:** An optimistic strategy whereby $\epsilon = 0.5$. Given (11), if PentoRob is the superstates of `taking` or `placing` then this is taken as sufficient evidence for showing commitment, and therefore confirmations or repairs can be interpreted during robotic movement.

The users rate the system after every round on a 5-point Likert scale questionnaire asking the questions (albeit in German) as shown in Fig. 5. We hypothesized that the fluid setting would be rated more favourably, due to its behaviour being closer to that observed in manipulator roles in human-human interaction. We had several objective criteria: an approximation to task success as the average time taken to place a piece in the correct location, and also as indications of the variety of dialogue behaviour the repair rate per word (i.e. words classified as belonging to a *repair* act) and the confirmation rate per word.

5.1 Results

Several rounds had to be discarded due to technical failure, leaving 24 ratings from the easier rounds (1 and 2) and 18 from the harder round 3. We found no significant differences in the overall questionnaire responses, however for the easier rounds alone, there was a significant preference for the Fluid system for the feeling that the system understood the user (Fluid mean=3.88, Incremental mean=3.18, Mann-Whitney U $p < 0.03$). The Fluid setting was not preferred significantly in terms of ease of playing ($p < 0.06$), and the ratings were generally positive for ratings of fun and wanting to play again but without significant differences between the two settings.

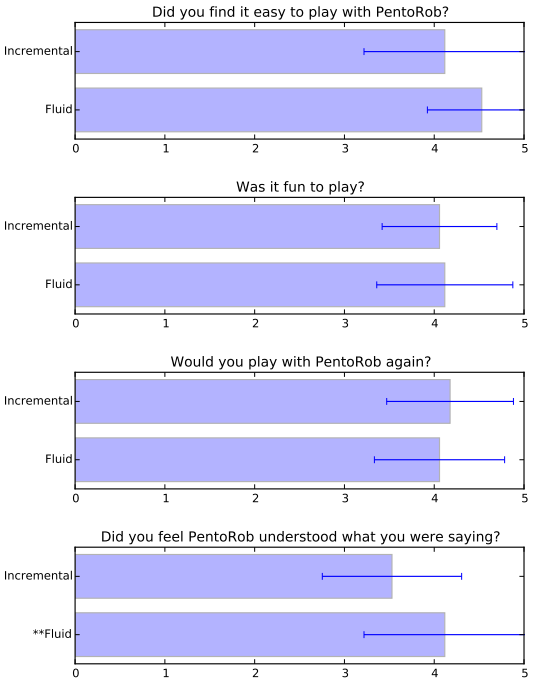

Figure 5: User ratings of the systems in the easier setting (** = Mann-Whitney U with $p < 0.05$)

Within the objective measures in terms of task success (time per piece placed), and rates of different incremental dialogue acts, there were no significant differences between the systems, only a tendency for a higher rate of confirmation words in the fluid setting. The limiting factor of the speed of the robotic arm meant the task success was not improved, however the noticeable increase in displaying understanding was likely due to the affordance of confirming and repairing during the robotic action.

5.2 Preliminary investigation into the User's criteria for showing commitment

For a preliminary investigation into the other parameter in our grounding model, we performed a study with 4 further participants who played with a system in both the modes described above again, but this time with δ, the $User$'s judgement of showing commitment to their goal (which is a confidence threshold for WAC's reference resolution hypothesis (8)) being set much lower– 0.05, compared to 0.2 in the first study. The lower threshold results in earlier, though possibly less accurate, reference resolution and consequent movement to target pieces.

We compared this group's objective measures to a random sample of 4 participants from the first

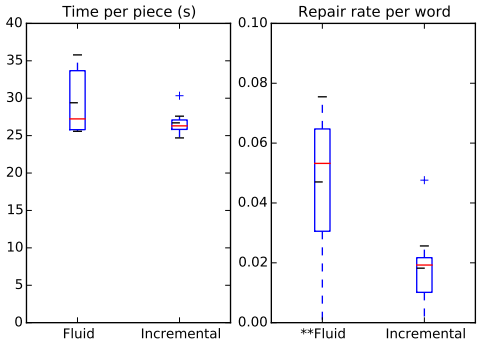

Figure 6: Preliminary result: Repair rates were significantly higher in the more fluid setting with a lower δ parameter of the grounding model whilst not affecting task success.

study, and there was a significant difference in repair rates (Fluid= 0.047 per word (st.d=0.024), Incremental=0.011 per word (st.d=0.011), T-test p <0.01) – see Fig. 6. Also, there was a tendency for higher rates of confirmation (Fluid= 0.245 per word (st.d=0.112), Incremental=0.151 per word (st.d=0.049), T-test p = 0.06). Encouragingly, the repair rates are in line with those reported in human-human similar task-oriented dialogue, with onsets occurring in 2-5% of words (Colman and Healey, 2011). However, also encouraging is that despite more time spent repairing and confirming in the more predictive system with the lower δ threshold, there was no effect on task success (e.g. see the near identical means for time taken to place each piece in Fig. 6).

5.3 Discussion

In the first experiment, the ratings results suggest the fluid setting's affordance of allowing confirmations and repairs during the robot's movement was noticed in easier rounds. More work is required to allow this effect to persist in the harder round, as severe failures in terms of task success cancelled the perception of fluidity.

The second experiment showed that the earlier movement of the robot arm to the target piece resulted in the user engaging in more repair of the movement, but this did not affect task success in terms of overall speed of completion. The degree to which the earlier demonstration of commitments to a goal during a user's speech, despite repair being required more often, can increase interactive success in more challenging reference situations will be investigated in future work.

6 Conclusion

We have presented a model of fluid, task action-based grounding, and have shown that it can be implemented in a robot that perceives and manipulates real-world objects. When general task-performance is good enough, the model leads to the perception of better understanding over a more standard incremental processing model.

There are some weaknesses with the current study. We intend to use more complex strength of evidence measures, for example for $Ev(UserGoal)$ using ASR hypotheses confidence thresholds (Williams, 2012), and having a more complex $Ev(RobotGoal)$ based on the robot's current position and velocity. We also want to explore learning and optimization for our incremental processing, with points of departure being (Paetzel et al., 2015), (Dethlefs et al., 2012), and the proposal by (Lemon and Eshghi, 2015).

The future challenge, yet potential strength, for our model is that unlike most approaches which assume a finite state Markov model for probabilistic estimation, we do not assume the Cartesian product of all possible substates needs to be modelled. The mathematics of how this can be done for a complex hierarchical model has had recent attention, for example in recent work in probabilistic Type Theory with Records (Cooper et al., 2014)– we intend to pursue such an approach in coming work.

Acknowledgments

We thank the three SigDial reviewers for their helpful comments. We thank Casey Kennington, Oliver Eickmeyer and Livia Dia for contributions to software and hardware, and Florian Steig and Gerdis Anderson for their help in running the experiments. 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.

Timo Baumann, Casey Kennington, Julian Hough, and David Schlangen. 2016. Recognising conversational speech: What an incremental asr should do

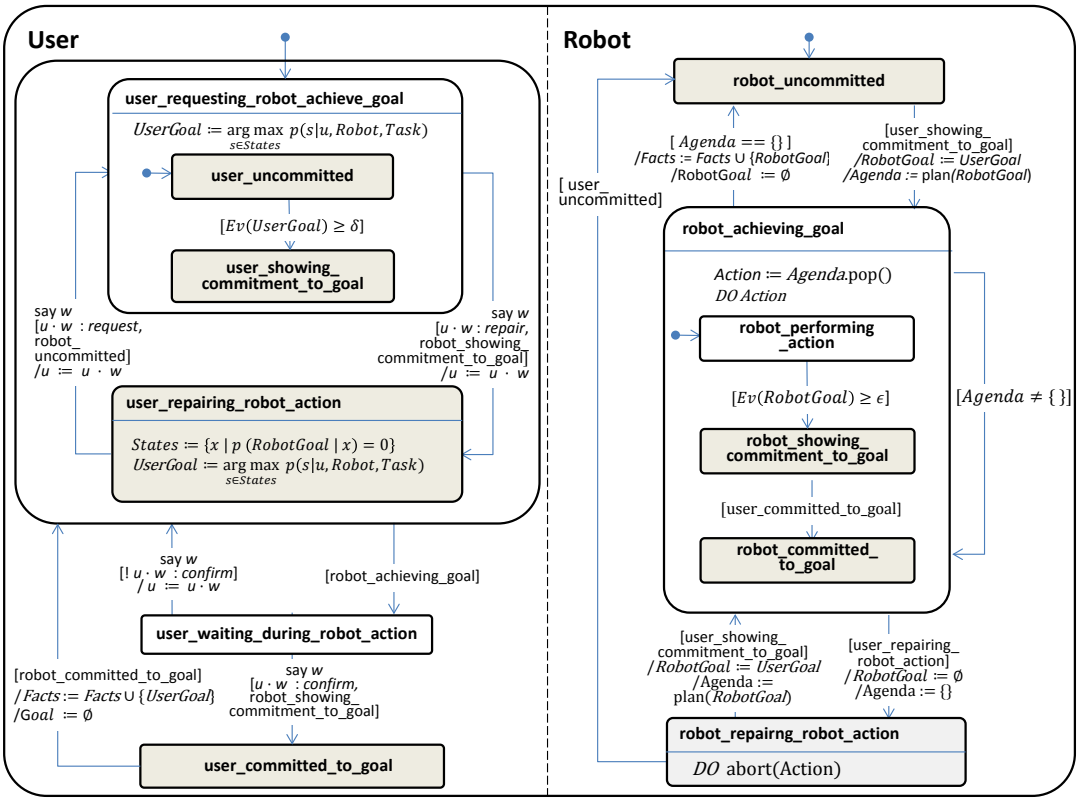

Figure 7: The full Interactive Statechart. States relevant for grounding are in grey.

for a dialogue system and how to get there. In *International Workshop on Dialogue Systems Technology (IWSDS) 2016*. Universität Hamburg.

Hendrik Buschmeier and Stefan Kopp. 2012. Using a bayesian model of the listener to unveil the dialogue information state. In *SemDial 2012: Proceedings of the 16th Workshop on the Semantics and Pragmatics of Dialogue*.

Okko Buß and David Schlangen. 2011. Dium – an incremental dialogue manager that can produce self-corrections. Proceedings of semdial 2011 (Los Angelogue).

Herbert H Clark and Susan E Brennan. 1991. Grounding in communication. *Perspectives on socially shared cognition*, 13(1991).

Herbert H Clark. 1996. *Using language*. Cambridge university press.

Marcus Colman and Patrick Healey. 2011. The distribution of repair in dialogue. In c. Hoelscher and T.F. Shipley, editors, *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, Boston, Massachussetts. Austinm TX:Cognitive Science Society.

Robin Cooper, Simon Dobnik, Shalom Lappin, and Staffan Larsson. 2014. A probabilistic rich type theory for semantic interpretation. In *Proceedings of the EACL Workshop on Type Theory and Natural Language Semantics (TTNLS)*, Gothenburg, Sweden. ACL.

Nina Dethlefs, Helen Hastie, Verena Rieser, and Oliver Lemon. 2012. Optimising incremental dialogue decisions using information density for interactive systems. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. ACL.

Arash Eshghi, Christine Howes, Eleni Gregoromichelaki, Julian Hough, and Matthew Purver. 2015. Feedback in conversation as incremental semantic update. In *Proceedings of the 11th International Conference on Computational Semantics*, London, UK. ACL.

Jonathan Ginzburg, Raquel Fernndez, and David Schlangen. 2014. Disfluencies as intra-utterance dialogue moves. *Semantics and Pragmatics*, 7(9).

Jonathan Ginzburg. 2012. *The Interactive Stance: Meaning for Conversation*. Oxford University Press.

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

Julian Hough, Iwan de Kok, David Schlangen, and Stefan Kopp. 2015. Timing and grounding in motor skill coaching interaction: Consequences for the information state. In *Proceedings of the 19th SemDial*

Workshop on the Semantics and Pragmatics of Dialogue (goDIAL), pages 86–94.

Nicholas R Jennings. 2001. An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4).

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.

Casey Kennington, Spyros Kousidis, Timo Baumann, Hendrik Buschmeier, Stefan Kopp, and David Schlangen. 2014. Better driving and recall when in-car information presentation uses situationally-aware incremental speech output generation. In *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM.

Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. 2010. Toward understanding natural language directions. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*. IEEE.

Geert-Jan M Kruijff. 2012. There is no common ground in human-robot interaction. In *Proceedings of SemDial 2012 (SeineDial): The 16th Workshop on the Semantics and Pragmatics of Dialogue*.

Oliver Lemon and Arash Eshghi. 2015. Deep reinforcement learning for constructing meaning by babbling. In *Interactive Meaning Construction A Workshop at IWCS 2015*.

Changsong Liu, Jacob Walker, and Joyce Y Chai. 2010. Ambiguities in spatial language understanding in situated human robot dialogue. In *AAAI Fall Symposium: Dialog with Robots*.

Changsong Liu, Rui Fang, and Joyce Y Chai. 2012. Towards mediating shared perceptual basis in situated dialogue. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. ACL.

Matthew Marge and Alexander I Rudnicky. 2011. Towards overcoming miscommunication in situated dialogue by asking questions. In *AAAI Fall Symposium: Building Representations of Common Ground with Intelligent Agents*.

Chris McKinstry, Rick Dale, and Michael J Spivey. 2008. Action dynamics reveal parallel competition in decision making. *Psychological Science*, 19(1):22–24.

Maike Paetzel, Ramesh Manuvinakurike, and David DeVault. 2015. so, which one is it? the effect of alternative incremental architectures in a high-performance game-playing agent. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

Julia Peltason and Britta Wrede. 2010. Pamini: A framework for assembling mixed-initiative human-robot interaction from generic interaction patterns. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. ACL.

Matthew Purver, Arash Eshghi, and Julian Hough. 2011. Incremental semantic construction in a dialogue system. In J. Bos and S. Pulman, editors, *Proceedings of the 9th IWCS*, Oxford, UK.

Antoine Raux and Mikio Nakano. 2010. The dynamics of action corrections in situated interaction. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. ACL.

Johan Schalkwyk, Doug Beeferman, Françoise Beaufays, Bill Byrne, Ciprian Chelba, Mike Cohen, Maryam Kamvar, and Brian Strope. 2010. Your Word is my Command: Google Search by Voice: A Case Study. In *Advances in Speech Recognition*. Springer.

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

Gabriel Skantze and Samer Al Moubayed. 2012. Iristk: a statechart-based toolkit for multi-party face-to-face interaction. In *Proceedings of the 14th ACM international conference on Multimodal interaction*. ACM.

Gabriel Skantze and Anna Hjalmarsson. 2010. Towards incremental speech generation in dialogue systems. In *Proceedings of the 11th Annual Meeting of SIGDIAL*. ACL.

Michael K Tanenhaus and Sarah Brown-Schmidt. 2008. Language processing in the natural world. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 363(1493):1105–1122.

David R Traum and Staffan Larsson. 2003. The information state approach to dialogue management. In *Current and new directions in discourse and dialogue*. Springer.

David R Traum. 1994. A computational theory of grounding in natural language conversation. Technical report, DTIC Document.

Jason D Williams. 2012. A belief tracking challenge task for spoken dialog systems. In *NAACL-HLT Workshop on Future Directions and Needs in the Spoken Dialog Community: Tools and Data*. ACL.

A Supplemental Material

The full statechart is in Figure 7.