Autonomous Robots (2019) 43:389–414
https://doi.org/10.1007/s10514-018-9795-5

Dempster-Shafer theoretic resolution of referential ambiguity

Tom Williams[1] · Fereshta Yazdani[3] · Prasanth Suresh[1] · Matthias Scheutz[2] · Michael Beetz[3]

Received: 30 November 2017 / Accepted: 2 August 2018 / Published online: 20 August 2018

Abstract
Robots designed to interact with humans in realistic environments must be able to handle uncertainty with respect to the identities and properties of the people, places, and things found in their environments. When humans refer to these entities using *under-specified* language, robots must often generate *clarification requests* to determine which entities were meant. In this paper, we first present recommendations for designers of robots needing to generate such requests. We then show how a Dempster-Shafer theoretic pragmatic reasoning component capable of generating requests to clarify *pragmatic* uncertainty can also generate requests to resolve *referential* uncertainty when integrated with probabilistic reference resolution and referring expression generation components. Our system is then demonstrated in a simulated alpine search and rescue context enabled by a novel hybrid architecture.

Keywords Human–Robot interaction · Human–robot dialogue · Natural language pragmatics · Dempster-Shafer theory · Alpine search and rescue

1 Introduction

Imagine a robot named Cindy and a human named Bob. Cindy and Bob are working together in a disaster relief scenario, and have just left a kitchen containing two medical kits: one on a table, and one on a counter. After driving for a few minutes, Bob turns to Cindy and asks "Can you go back to the kitchen and grab the medical kit?"

This is one of several papers published in *Autonomous Robots* comprising the "Special Issue on Robotics Science and Systems".

✉ Tom Williams
twilliams@mines.edu

Fereshta Yazdani
yazdani@cs.uni-bremen.de

Prasanth Suresh
psengadusuresh@mines.edu

Matthias Scheutz
matthias.scheutz@tufts.edu

Michael Beetz
beetz@cs.uni-bremen.de

[1] MIRRORLab, Colorado School of Mines, Golden, CO, USA

[2] Human-Robot Interaction Laboratory, Tufts University, Medford, MA, USA

[3] Institute for Artificial Intelligence, Universität Bremen, Bremen, Germany

To successfully fulfill Bob's request, Cindy must resolve two types of ambiguity. Bob's request is *pragmatically ambiguous* as it could be interpreted *directly* (as a literal question as to Cindy's abilities) or *indirectly* (as a command to Cindy). Bob's request is *referentially ambiguous* because it could refer to either the medical kit on the table or the one on the counter. Referential ambiguity is a key challenge in realistic robotics applications, as in realistic task contexts there may be a large number of task-relevant entities that could be referred to. What is more, these entities must be disambiguated using the properties they hold and the relationships between themselves: a challenge not often encountered in other conversational applications such as personal assistants, where many candidate referents (e.g., people, songs, and commercial locations) have uniquely identifying proper names. This challenge is made even more difficult by the necessity for grounding and uncertainty and incompleteness of knowledge in realistic robotics applications (Mavridis 2015).

When humans are confronted with this sort of ambiguity, they typically resolve it using *clarification requests* such as "Do you want me to retrieve the medical kit that is on the counter or the medical kit that is on the table?" (Tenbrink et al. 2010). In previous work, we showed how Dempster-Shafer (DS)-theoretic pragmatic reasoning could be used to both identify sources of pragmatic ambiguity and generate pragmatically appropriate clarification requests (Williams et al.

2015) to resolve such ambiguity. However, that work could not resolve *referential* ambiguity, and assumed that information about all referents was stored in a single, centrally located knowledge base (cf. Williams and Scheutz 2016a).

In this work, we demonstrate the integration of a DS-theoretic pragmatic reasoning component with algorithms for performing reference resolution and referring expression generation under uncertainty, and show how this integration allows a robot to identify, and generate clarification requests to resolve, *referential* ambiguity as well. This approach is uniquely tailored to human–robot interaction (HRI) contexts, as it produces human-preferred clarification requests that conform with the pragmatics of human–robot dialogue.

The remainder of this paper [which is largely based on our previous work in (Williams and Scheutz 2017b)] proceeds as follows: First, we discuss previous work on clarification request generation in human–robot interaction contexts. Next, we present the results of a human-subjects experiment in which previous findings regarding human preferences with respect to robot clarification request formulation are replicated and refined. We then present an HRI-oriented framework for clarification request generation along with an algorithmic implementation of that framework which is designed to align with human preferences. As part of this evaluation, we introduce a novel hybrid architecture, in which the language understanding and generation components of the ADE-based DIARC architecture described in this paper are integrated with the ROS-based KnowRob Knowledge processing framework and CRAM architecture. Here, KnowRob+CRAM provide knowledge of—and potential for action within—a simulated alpine search and rescue context. Using this hybrid architecture, we demonstrate the behavior of our presented approach within that alpine search and rescue context. Finally, we conclude with possible directions for future work.

2 Background

In this section, we first discuss previous work on natural language generation in general, and clarification request generation. We then critique that work in order to generate a set of hypotheses regarding human preferences that should be accounted for when designing architectural components for language-capable robots.

There has been much previous work in developing general natural language generation (NLG) systems. For example, Reiter et al. present an NLG framework comprised of six stages: content determination, document structuring, aggregation, lexical choice, referring expression generation (REG), and realization (Reiter et al. 2000). It is unclear, however, whether such frameworks are well suited to *situated contexts* in which an agent is embedded in a complex, dynamic, environment rife with uncertainty and ambiguity (Matarić 2002). In HRI, for example, NLG is often performed to *solicit* information, whereas in non-situated contexts it is more typically performed to *provide* information. Accordingly, we propose an *HRI-oriented* clarification request generation framework comprised of six stages: (1) uncertainty identification, (2) decision to communicate, (3) utterance choice, (4) content selection, (5) surface realization, and (6) speech synthesis. This framework extends the framework introduced in previous work (Williams and Scheutz 2016b) by re-introducing a stage dedicated to content selection. In this paper, we will discuss this framework in detail, and then present an integrated approach that implements all six stages.

Given the role of clarification (also known as *repair*) as a central component of natural language dialogue (Ginzburg 2009; Schegloff 1987), clarification request generation itself has attracted a large amount of research overall (Benotti and Blackburn 2017; Gabsdil 2003; Purver et al. 2003; Rodríguez and Schlangen 2004; Stoyanchev et al. 2013; Traum 1994). However, there has been relatively little work on clarification request generation in *situated* contexts such as human–robot interaction. Recently, some researchers have used information-theoretic techniques to identify random variables which could have their entropy reduced if asked about. In such work, clarification requests have taken the form of yes/no questions about the properties of an object (Deits et al. 2013; Hemachandra et al. 2014; Purver 2004) or generic WH-questions (e.g., "What do the words *X* refer to?") (Tellex et al. 2013; Purver 2004).[1]

Recent experimental evidence from Marge and Rudnicky (2015) suggests, however, that in HRI contexts, people prefer robots to list multiple options rather than asking for confirmation about a single referent with a yes/no- or generic WH-question (cf. Clark 1996). This is particularly striking as the evidence suggests that people maintain this preference even when a yes/no- or generic WH-question would be more efficient (cf. Hemachandra et al. 2014).

In contrast, Kruijff et al. present an approach in which robots can generate multiple-option clarification requests such as "Do you mean the blue or the red mug, Anne?" through a *continual planning* approach (Kruijff et al. 2008) (see also Brenner and Kruijff-Korbayová 2008; Kruijff et al. 2006b). This approach, however, does not appear to be able to account for social context, uncertainty, or ignorance, and is only used for generation. The ability to handle context (Benotti and Blackburn 2017), especially social context, is crucial for enabling natural HRI (Mutlu and Forlizzi 2008), and typical HRI scenarios are plagued by uncertainty and

[1] While not directly relevant to the present work, there has also been research on using interaction patterns to identify opportunities for clarification in situated settings (Carrillo and Topp 2016).

ignorance (Talamadupula et al. 2011). An eldercare robot, for example, is not likely to be familiar with every object in the home of the elder it is assisting, nor with every person who might be referred to. Furthermore, the robot is unlikely to have *uncertainty-free* knowledge of all of the properties and relations involving those entities it *does* know of. This inability to handle uncertainty is shared by similar ontology-based approaches to disambiguation (Maurtua et al. 2016).

3 Experiment one: preference assessment

In developing a new HRI-oriented approach to clarification request generation, our goal is to account for uncertainty and ignorance while taking *human preferences* into account. We believe that the previous work discussed thus far has not adequately considered what type of utterances humans *prefer* to use and be used. We hypothesize that there are three categories of human preferences that should affect the design decisions made when developing HRI-oriented clarification request generation algorithms. In this section, we first describe these three hypotheses, and then present the results of a human subjects experiment designed to test them.

3.1 Design hypotheses

3.1.1 Presentation of options

Marge and Rudnicky (2015) suggests people prefer that robots list options rather than ask yes/no- or generic WH-questions. But clearly there are limits to this preference. If a robot is asked "Could you get me some ice cream?", it is unlikely that humans will prefer the robot to list twenty-seven available flavors instead of just asking "Which flavor would you like?". It is not yet clear, however, how many options can be listed until the use of a list is no longer preferable. We hypothesize **(H1)** that humans prefer options to be listed *only for a very small number of options*.

3.1.2 Demonstration of intention understanding

Similarly, many previous approaches use clarification requests that do not demonstrate understanding of the *meaning* of the sentence. If a robot is asked "Could you get me some ice cream," a robot that replies "What do the words 'ice cream' refer to" or "Do you mean 'the chocolate ice cream' or 'the vanilla ice cream"' does not allow its interlocutor to discern whether their *intention* was understood. In contrast, a robot that replies "*Would you like me to get you* the chocolate ice cream or the vanilla ice cream?" communicates understanding that the human wants ice cream *brought to them*. We hypothesize **(H2)** that humans prefer clarification requests that demonstrate understanding of their intentions.

3.1.3 Pragmatic appropriateness

Finally, a robot that *does* generate clarification requests reflecting its understanding of human intentions will almost certainly need to use *indirect speech acts* (Searle 1975) (e.g., *Would you like* me to get you the chocolate ice cream or the vanilla ice cream?), as the direct alternatives (e.g., "I have an intention to know whether you want me to have a goal to bring you the chocolate ice cream or the vanilla ice cream") are hard to express succinctly, and are viewed as less polite. We hypothesize **(H3)** that humans prefer indirectly rather than directly phrased clarification requests.

3.2 Methodology

Participants were recruited (20 Male, 10 Female) using Amazon Mechanical Turk. All participants were American, and ranged in age from 24 to 48 (M = 32.67, SD = 6.30). Only high-reputation participants were used to guard against potential participation from automated "bots". Each participant was asked seven simple questions, presented in a randomized order. Participants were told to imagine commanding a robot to "pick up the mug" in a scenario with several different-colored mugs on a table. For each question (which differed in the number of candidate mugs) two ways of asking for clarification were presented. Participants were asked to indicate which option they would prefer the robot to use. Finally, participants answered an "attention check" question asking them to indicate what object was talked about in the previous questions, in order to guard against participants "clicking through" questions without reading them. No participants failed this attention check.

The first five questions evaluated **H1**. In each case, participants chose between an option that listed out all options (ranging from "Would you like the red mug or the orange mug?" to "Would you like the red mug or the orange mug or the yellow mug or the green mug or the blue mug or the purple mug?") and a catch-all ("Which mug would you like?"). For each question, the number of options provided directly corresponded to the number of entities in the context description provided to participants for that question.

The sixth question evaluated **H2**. Participants chose between an option that indicated understanding of the speaker's goals ("Would you like the red mug or the green mug?") and one that did not ("Do you mean the red mug or the green mug?").

The last question evaluated **H3**. Participants chose between a pragmatically appropriate option ("Would you like the red mug or the blue mug?") and a pragmatically inappropriate option ("I have an intention to know if you want me to have a goal to bring you the red mug or the blue mug.").

These seven questions were presented in a randomized order to each participant, and the options for each question

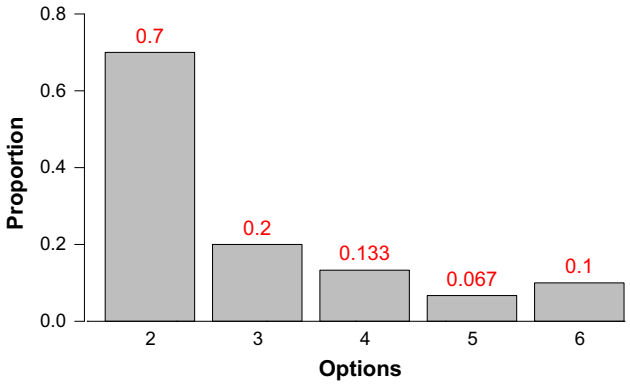

Fig. 1 *Experiment one results*. Proportion of participants who preferred options to be listed, for each candidate number of options

were also presented in a randomized order for each participant.

Before moving on, we must briefly discuss our inclusion of this sentence. It seems obvious to us that no reasonable participant would prefer this tortured form, and thus do not expect the statistical results for this particular question to be terribly surprising. However, we included this question in our experiment to call attention to the obvious shortcomings of such forms, because this calls attention to the importance of *pragmatic* natural language generation. This form directly and literally communicates the robot's intentions; accordingly, if a robot does not have the mechanisms necessary to generate pragmatically appropriate language, it would have no choice but to use that direct (and clearly suboptimal) form.

3.3 Results

As shown in Fig. 1, our results show that 70% of participants preferred options to be listed when there were only two options. But for more than two options, this number rapidly shrank. Only 20% of participants preferred options to be listed when there were three options, and preference for listing all options fell lower still when more options were listed. This confirms but clarifies the previous findings of Marge and Rudnicky (2015), and suggests that robots likely do not need mechanisms for listing more than two options when there is referential ambiguity **(H1)**. Our results show that 80% of participants preferred the option that indicated understanding of their goals, supporting **H2**. Our results show that 93% of participants preferred the pragmatically appropriate option, supporting **H3**.

3.4 Discussion

The results of this experiment suggest three design recommendations. **(D1)** When phrasing clarification requests, if there are only two options, robots should present both options. Otherwise, robots should use a yes/no- or generic WH-question.[2] **(D2)** When phrasing clarification requests, robots should use phrasings that indicate that they understand the goals of their interlocutors. **(D3)** When phrasing clarification requests, robots should use pragmatically appropriate phrasings.

The experiment presented in this section presents an initial assessment of human preferences regarding clarification requests, but suffer from a number of limitations.

First, there are limitations with respect to stimuli. While we intended to evaluate communication of intentions and pragmatic appropriateness separately, these are multifaceted and interacting dimensions. Our utterance "Would you like the red mug or the green mug", for example, certainly communicates intentions more clearly than "Do you mean the red mug or the green mug?", but it could also be considered as more polite, and thus, potentially more pragmatically appropriate. Similarly, it is possible that participants may have been subtly affected by the use of multiple or-clauses. Future work will be needed to tease apart the difference between these two factors.

Second, there are limitations with respect to our prompt. Participants were asked to indicate which option they would prefer the robot to use, but this may also be multifaceted, with different people using different preference metrics. Future work will need to tease apart the differences between such metrics and their relative importance.

Third, there are limitations with respect to experimental setting: in crowdsourced experiments, it is not possible to control participants' experimental setting, and we did not control for web browser, operating system, or other factors that may have impacted viewing experience. In addition, while attention checks are helpful, it does seem strange that two participants out of thirty chose what would seem to be the clearly less preferable choice for the last question. If these two participants had been treated as outliers and removed, we would have had the following results: For two-through-six presented options, 67.9, 17.9, 10.7, 3.5, and 7.1 percent, respectively, of the remaining participants would have preferred options to be listed, and 78.6 percent of the remaining participants would have preferred the request that demonstrated understanding of intentions.

Fourth, there are limitations with respect to context generalizability. Our results are specific to environments in which there are between two and six objects of the same type, assumed to be visible to both speaker and listener. It is important to recognize that under variations of these contextual factors, human preferences may have changed. For example, when two agents do not share an environment, the hearer asking for clarification may need to clarify the spe-

[2] Future research will be needed to determine how the *content* of the options to be offered may impact how this decision is made. The results of such research may suggest refinements of this recommendation.

cific candidates they see, or the fact that they see a number of candidates, before asking for clarification. In such contexts, people may actually prefer a greater number of options to be listed because it is inherently informative to them. In future work, it will be important to reevaluate human preferences in a wider range of scenarios, under varying levels of shared knowledge. For the time being, we will simply conclude that, in contexts such as those we evaluated, listing options for two candidates and generating a more general question for more than two candidates is a reasonable strategy.

In the following sections, we will demonstrate how the integration of architectural components for reference resolution and pragmatic reasoning facilitates an approach to clarification request generation that not only fulfills all three of these design decisions, but also satisfies capabilities missing from previous approaches (e.g., context sensitivity, handling of uncertainty and ignorance, and use for both understanding and generation). In Sect. 4, we will begin by defining a general, HRI-oriented framework for clarification request generation. In Sect. 5, we then present our algorithms and architectural mechanisms for implementing each stage of that framework.

4 An HRI-oriented framework for clarification request generation

We identify six stages necessary for successful clarification request generation: (1) uncertainty identification, (2) decision to communicate, (3) utterance choice, (4) content selection, (5) surface realization, and (6) speech synthesis. In this section we describe the actions necessary at each stage.

4.1 Uncertainty identification

Suppose that in our original example, Bob had asked Cindy "Can you grab the medkit?" During the stage of *uncertainty identification*, The robot (Cindy) must determine if it is unsure how to interpret any part of this utterance. This may be uncertainty as to what entities are being *referenced*, e.g., *which* medkit Bob is referring to, or uncertainty as to the speaker's *intentions*, e.g., whether Bob wishes Cindy to bring him the medkit or whether he meant something else by the utterance. Furthermore, this uncertainty may take different forms (cf. Stirling 2010): the utterance may be *ambiguous* (e.g., if Cindy knows of multiple medkits) or the utterance may reveal *ignorance* (e.g., if Cindy knows of no medkits, or is unsure whether a particular object qualifies as a "medkit").

4.2 Decision to communicate

If a robot has identified a point in need of clarification, it must decide whether it would be appropriate to actually ask for clarification. This decision will depend on a variety of factors: Is it permissible for the robot to ask for clarification? Is the robot's interlocutor likely to be able to provide clarification? Would obtaining clarification really be the highest utility action at the current time (compared to, e.g., exploration)? For example, if Cindy determines there are actually two medkits that Bob could be referring to, but while coming to this decision Bob has already engaged another teammate in conversation, it may be necessary for Cindy to wait until this conversation finishes before asking for clarification.

4.3 Utterance choice

Once a robot has decided to request clarification on a particular point, it must determine what utterance form to use to communicate its request: depending on the relationship between the robot and its interlocutor, and the obligations of each party, certain utterance forms may be more or less appropriate (Brown 1987). For example, if Cindy is Bob's subordinate, it may be more appropriate to use an *indirect request* such as "Which medkit do you need?", whereas if Cindy is Bob's superior, it may be more appropriate to use a *direct request* such as "Tell me which medkit you need."

This processing step is not typically included in traditional NLG frameworks, which do not typically need to account for social context or dialogue context. They instead typically include a *document structuring* stage, where the agent determines the order in which to convey multiple utterances (Reiter et al. 2000). Because clarification request generation in HRI *typically* only involves a single utterance, we do not currently handle this step, but it will be an important topic for future work. A robot may, for example, need to preface a clarification request by stating what parts of an utterance it *did* understand.

4.4 Content selection

Once a robot chooses an utterance form to use, it must determine a referring form to use for each referent within that utterance. If the robot elects to generate a definite noun phrase, it must choose what properties to use to describe entities referenced in that utterance (Garoufi and Koller 2014). For example, if the robot decides to use an utterance of the form "Would you like [medkit$_1$]", it must choose how to actually describe medkit$_1$, e.g., by referring to it by its location, color, size, and so forth. Alternatively, at this stage the robot may elect to generate an anaphoric or deictic expression (involving eye gaze, gesture, or other modalities) rather than a complex noun phrase.

4.5 Surface realization

Once a robot chooses a referring form to use, it must decide what words to use in service of that referring form (as well

as how to communicate information from other modalities, if applicable), especially in order to express properties selected during content selection, in the case of complex noun phrases.

4.6 Speech and gesture synthesis

Finally, once a robot determines what words to use, it must synthesize an appropriate sound pattern and/or execute appropriate communicative motor programs.

5 Algorithmic approach

In this section, we will present the set of algorithms and architectural mechanisms we have developed in order to implement each stage of the framework defined in the previous section. While this framework is designed to incorporate both language and gesture, we will focus on the linguistic portions of the framework in this work, leaving gestural communication to future work. Throughout this section, we will refer to various components of the Distributed Integrated Affect Reflection Cognition (DIARC) architecture (Schermerhorn et al. 2006), in which these algorithms and mechanisms are implemented. A diagram of these components and their connections can be seen in Fig. 2.

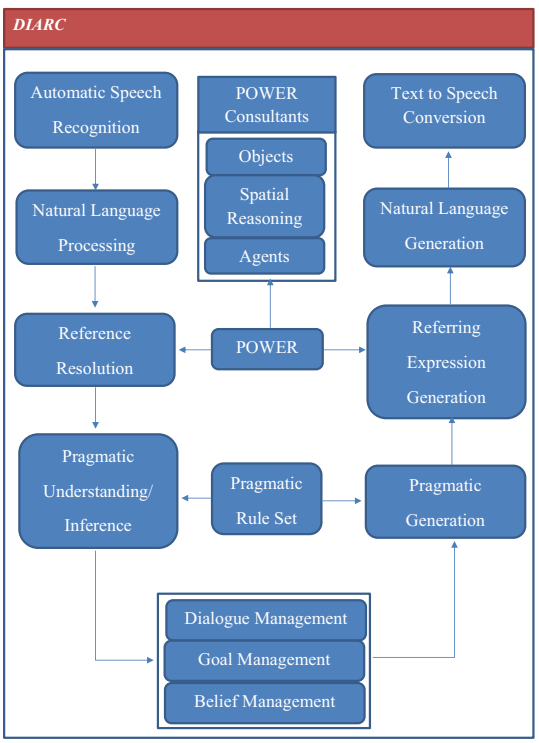

Fig. 2 Architectural diagram of the DIARC architecture

Notation[3]

M	A robot's *world model* of entities $\{e_0 \ldots e_n\}$.
Λ	A set of logical formulae $\lambda_0 \ldots \lambda_n$, denoting (literal, direct) semantic *connotation* of an incoming utterance.
V	A set of free variables found in Λ.
Γ	A set of bindings from variables in V to entities in M, denoting the semantic *denotation* of an incoming utterance.
Φ	A *satisfaction* variable which is *True* iff all formulae in Λ *hold* when bound using Γ.

5.1 Uncertainty identification

The first step in our clarification request generation framework is identifying whether or not there is uncertainty that needs to be clarified. To achieve this, we first determine the set of referential candidates and their respective levels of uncertainty. We then provide those candidates to a pragmatic inference component which produces a set of uncertain candidate interpretations. In this section, we will detail this process and the integration challenges it presents.

5.1.1 Reference resolution

Our approach uses the DIST- POWER framework to facilitate access to information about entities a robot knows of (Williams and Scheutz 2016a). The DIST- POWER framework uses a set of "Consultants" to integrate a central, domain-independent open-world reference resolution component with a set of heterogeneous knowledge bases distributed throughout a robot architecture, potentially residing on multiple machines (Williams 2017a). In our instantiation of this framework, we make use of GH- POWER: our *Givenness Hierarchy*-theoretic reference resolution algorithm (Williams et al. 2016). Based on the theoretical linguistic framework presented by Gundel et al. (1993), GH- POWER treats DIST- POWER's distributed memory system as a Long Term Memory Store, and builds on top of it a set of hierarchical caches representing models of the robot's Discourse Context, Short-Term Memory, and Focus of Attention, as shown in Fig. 3. This allows GH- POWER to resolve a wide array of referring expressions (REs). And, like the non-GH-theoretic version of POWER, GH- POWER handles both uncertain and open worlds. Finally, GH- POWER offers a significant performance advantage over the use of DIST- POWER

[3] We have chosen to adopt (and extend) the notation used in Tellex et al. (2011) in order to facilitate easier comparison to related work. We would advocate for its adoption as a common notation across the reference resolution and symbol grounding communities.

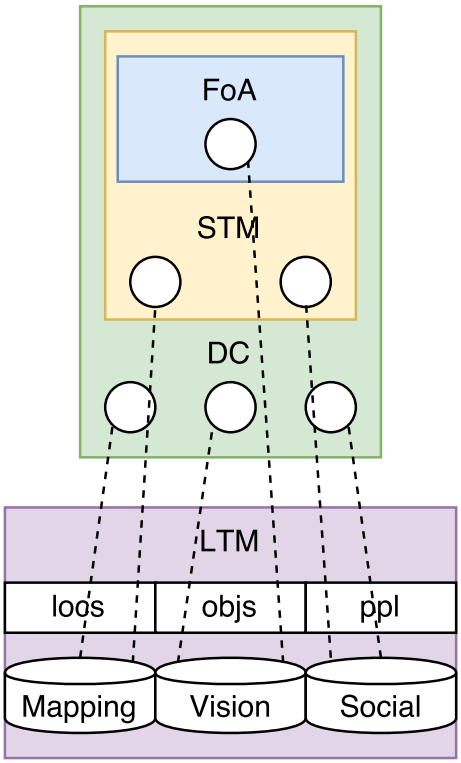

Fig. 3 *Memory model* The Focus of Attention, Short Term Memory, Discourse Context are hierarchically nested, and contain references to entities stored in the Distributed, Heterogeneous Knowledge Bases which comprise long-term memory (*Mapping, Vision, Social*), access to which is enabled and controlled using a set of Consultants (*locs, objs, ppl*)

by itself. DIST- POWER has worst-case time complexity of $O(j * m^k)$ assuming a query of j k-arity predicates[4] and a world model containing m entities (Williams 2017b). While GH- POWER uses DIST- POWER in the worst case to resolve definite noun phrases, in most cases this will be unnecessary, as entities already being discussed in a conversation are likely to be identified in, e.g., the small set of currently activated entities, precluding the need for an expensive search through long-term memory. For the sake of simplicity, we will use POWER to refer to the distributed, GH-theoretic form of the POWER algorithm and its associated data structures.

Like many others (see, e.g. Kollar et al. 2017; Lemaignan et al. 2017; Matuszek et al. 2012; Tenorth and Beetz 2017; Zettlemoyer and Collins 2012), we use an approach where natural language semantics are represented using a simple, structured, predicate logic extracted from text using a CCG or Dependency Parser. While the logical forms we use are not sufficiently flexible to handle the variety of temporally complex expressions that can be handled using, e.g., Temporal or Dynamic Logics (cf. Dzifcak et al. 2009), they *are* able to sufficiently represent the referring language found in the types of clarification dialogues discussed in the paper. Moreover, they provide an intermediary representation that allows discussion and reasoning that is not possible in approaches that directly and statistically associate vision and language.

POWER uses the logical form of a referring expression to (1) hypothesize new representations for previously unknown referents, and (2) produce a distribution $P(\Phi \mid \Gamma, \Lambda)$; that is, the probability of successful satisfaction conditioned on binding hypotheses from variables to *known* referents:

$$\{\Gamma_0 = \{\gamma_{0_0} \ldots \gamma_{0_n}\} \ldots, \Gamma_e = \{\gamma_{e_0} \ldots \gamma_{e_n}\}\}$$
and semantic parse hypotheses:
$$\{\Lambda_0 = \{\lambda_{0_0} \ldots \lambda_{0_n}\} \ldots, \Lambda_e = \{\lambda_{e_0} \ldots \lambda_{e_n}\}\}.$$

For example, suppose Bob asked Cindy "Can you grab the medical kit?" Cindy may parse this into something like

$$QuestionYN(b, s, can(s, grab(s, X)))$$

with additional semantic content $\Lambda_i = \{medkit(X)\}$ (Hereafter, we will use the abbreviations b="bob" and s="self"). If Cindy is 70% sure that the e_5 is a medical kit, reference resolution will produce:

$$P(\Phi = True \mid \Gamma = \{X \rightarrow e_5\}, \Lambda = \{medkit(X)\}) = 0.7$$

All sufficiently probable referential hypotheses are then used to create a set of *bound utterances with supplemental semantics* (BUSSes) $\Psi = \{\psi_0 \ldots \psi_n\}$. Each $\psi_i \in \Psi$ is associated with a unique sufficiently probable binding γ_i from variables found in the parsed utterance form and its supplemental semantics to entities found in Long Term Memory. For example, the BUSS associated with form $QuestionYN(b, s, can(s, grab(s, X)))$, semantics $\{medkit(X)\}$, and binding $\{X \rightarrow e_5\}$ would be:

$$\{QuestionYN(b, s, can(s, grab(s, e_5))) \wedge medkit(e_5)\}.$$

One could then create a *distribution* over BUSSes $P(\psi_i) = P(\Gamma_i, \Lambda_i \mid \Phi_i)$ using Bayes' Rule, *if* the next natural language component used a Bayesian framework. In fact, the next component in our architecture (i.e., the pragmatic reasoning component) uses a more general *Dempster-Shafer (DS) theoretic* framework (for reasons described in the next section), and thus another approach must be taken (Williams et al. 2015).

5.1.2 Dempster-shafer theory: a primer

Dempster-Shafer (DS) Theory is a generalization of the Bayesian uncertainty framework that allows for elegant

[4] Note that for most utterances j will be very small, and k will in almost all circumstances be either 1 or 2.

reasoning about uncertainty and ignorance even when distributional information is unavailable (Shafer 1976). DS Theory is an attractive option for HRI domains in which agents may encounter new entities and concepts only a small number of times, with no information regarding the distribution underlying their occurrence. DS Theory is also useful for tasks such as pragmatic reasoning, because it would be impractical to store priors over all combinations of intentions and contexts, as would be required in a Bayesian framework; and because it allows the use of DS-based logical rules such as Modus Ponens, which cannot be used in a strictly Bayesian framework (Núñez et al. 2013b, a; Tang et al. 2012).

In Dempster-Shafer Theory, the uncertainty of an event E is represented using the interval $[Bl(E), Pl(E)]$. $Bl(E)$ and $Pl(E)$ are the *belief* and *plausibility* of E: lower and upper bounds on $P(E)$ such that $0 \leq Bl(E) \leq P(E) \leq Pl(E) \leq 1$. The *width* of this uncertainty interval $(Pl(E) - Bl(E))$ indicates the degree of *ignorance* in event E. For a set of mutually exclusive singleton hypotheses Θ (known as a *Frame of Discernment (FoD)*), a *basic belief assignment (BBA)* $m(\cdot) : 2^{\Theta} \rightarrow [0, 1]$ can be defined, assigning a probability mass to each set in the power set of hypotheses. Another, potentially more intuitive way of viewing DS-Theory, presented by Dempster (2008), is to view a *Dempster-Shafer Model* (i.e., the Body of Evidence) as an assignment from each assertion you could make about a given *State Space Model* (each corresponding to a one of the 2^n possible subsets of that State Space Model (i.e., Frame of Discernment)) to a triple (p, q, r), where p is the probability "for" the assertion, q is the probability "against" the assertion, and r is the probability of "don't know". Below, we provide more formal definitions for each of DS Theory's basic notions.

Frame of Discernment

In DS theory, a set of elementary events of interest is called a *Frame of Discernment* (FoD). A FoD is a finite set of mutually exclusive events $\Theta = \{\theta_1, ..., \theta_N\}$. The power set of Θ is denoted by $2^{\Theta} = \{A : A \subseteq \Theta\}$.

Basic Belief Assignment

Each set $A \in 2^{\Theta}$ has a certain weight, or *mass* associated with it. A *Basic Belief Assignment* (BBA) is a mapping $m_{\Theta}(\cdot) : 2^{\Theta} \rightarrow [0, 1]$ such that $\sum_{A \subseteq \Theta} m_{\Theta}(A) = 1$ and $m_{\Theta}(\emptyset) = 0$. The BBA measures the support assigned to the propositions $A \subseteq \Theta$ only. The subsets of A with non-zero mass are referred to as *focal elements* and comprise the set F_{Θ}. The triple $E = \{\Theta, F_{\Theta}, m_{\Theta}(\cdot)\}$ is called the *Body of Evidence* (BoE).

Belief, Plausibility, and Uncertainty

Given a BoE $\{\Theta, F_{\Theta}, m_{\Theta}(\cdot)\}$, the *belief* for a set of hypotheses A is $Bel(A) = \sum_{B \subseteq A} m_{\Theta}(B)$. This belief function captures the total support that can be committed to A without also committing it to the complement A^c of A. The *plausibility* of A is $Pl(A) = 1 - Bel(A^c)$. Thus, $Pl(A)$ corresponds to the total belief that does not contradict A. The *uncertainty* interval of A is $[Bel(A), Pl(A)]$, which contains the true probability $P(A)$. In the limit case with no uncertainty, we get $Pl(A) = Bel(A) = P(A)$.

Thus, we see how information regarding the probability of some event can be gathered from the Dempster-Shafer theoretic notions of belief and plausibility, and how these notions themselves are derived from the masses m_{Θ} ascribed to specific hypotheses.

5.1.3 Integration of bayesian and dempster-shafer theoretic architectural components

In this section, we will discuss how the set of BUSSes constructed in Sect. 5.1.1 is transformed into a DS-theoretic Body of Evidence for use by our pragmatic reasoning module. Recall that the output of reference resolution was a set $\Psi = \{\psi_0 \ldots \psi_n\}$ where each ψ_i was a *Bound Utterance with Supplemental Semantics (BUSS)* with associated likelihood $P(\Phi_i \mid \Gamma_i, \Lambda_i)$, reflecting the degree to which the chosen variable bindings Γ_i are believed to satisfy the logical constraints specified in the set of supplemental semantics Λ_i, based on the probability judgments provided by POWER's Consultants.

Using this set of BUSSes, we define a Body of Evidence with Frame of Discernment $\Theta = \{\theta_1, \ldots, \theta_n\}$ where each θ_i is a mutually exclusive singleton hypothesis described by BUSS ψ_i, and with a Basic Belief Assignment assigning each θ_i the following mass:

$$\frac{P(\Phi_i \mid \Gamma_i, \Lambda_i)}{\sum_{j=0}^{|\Theta|} P(\Phi_j \mid \Gamma_j, \Lambda_j)} \tag{1}$$

This operation is linear in the number of referential hypotheses. Note, here, that these singleton hypotheses are guaranteed to be mutually exclusive because each corresponds to a particular set of intended referents: POWER only provides probability judgments about specific referential hypotheses, and does not provide any evidence for sets of hypotheses, which is why no mass is assigned to non-singleton sets. As mass is only assigned to singleton sets, this means that:

$$Bl(\theta_i) = Pl(\theta_i) = m(\theta_i).$$

The confidence interval associated with each hypothesis according to this mass assignment is identical to:

$$[Bl(\Gamma_i, \Lambda_i \mid \Phi_i), Pl(\Gamma_i, \Lambda_i \mid \Phi_i)]$$

as calculated using Heendeni et al. (2016)'s DS-theoretic equivalent to the formulation of Bayes' Rule (Eq. 2) when a

uniform prior distribution $Bl(\Gamma, \Lambda) = Pl(\Gamma, \Lambda) = \frac{1}{|\Theta|}$ is assumed.

$$\begin{aligned} Bl(A|B) &\geq \frac{Bl(B|A)Bl(A)}{Bl(B|A)Bl(A) + Pl(B|\bar{A})Pl(\bar{A})} \\ Pl(A|B) &\leq \frac{Pl(B|A)Pl(A)}{Pl(B|A)Pl(A) + Bl(B|\bar{A})Bl(\bar{A})} \end{aligned} \tag{2}$$

Here, Bl(A) and Pl(A) denote belief and plausibility as defined above, $\bar{A}$ denotes singleton hypotheses *not* appearing in the set A, and $Bl(A|B)$ and $Pl(A|B)$ denote *conditional* beliefs and plausibilities defined according to the Fagin-Halpern (FH) Conditional (Fagin and Halpern 1991).

For example, suppose speaker Bob asks "Can you grab the medkit that is near the book?", in an environment containing two medkits (e_1 and e_4) and two books (e_2 and e_3), and that POWER produces the following BUSSes, each with the same utterance form but describing a different referential hypothesis:

$$\begin{aligned} \psi_1 =& (QuestionYN(b, s, can(s, grab(s, e_1))) \\ & \wedge medkit(e_1) \wedge book(e_2) \wedge near(e_1, e_2)) \\ \psi_2 =& (QuestionYN(b, s, can(s, grab(s, e_1))) \\ & \wedge medkit(e_1) \wedge book(e_3) \wedge near(e_1, e_3)) \\ \psi_3 =& (QuestionYN(b, s, can(s, grab(s, e_4))) \\ & \wedge medkit(e_4) \wedge book(e_2) \wedge near(e_4, e_2)) \end{aligned}$$

Suppose the likelihoods associated by POWER with these BUSSes are:

$$\{\psi_1 \rightarrow 0.4, \psi_2 \rightarrow 0.6, \psi_3 \rightarrow 1.0\}.$$

The likelihoods used in this section are chosen arbitrarily for the sake of a clean working example, but are in practice derived by POWER from a variety of sources. For this example, the Consultant responsible for answering questions about these particular entities could determine using standard object recognition classifiers (e.g. Redmon et al. 2016) and spatial reasoning algorithms that it is 100% sure that e_4 is a medical kit, e_2 is a book, and e_4 qualifies as "near" e_2. Similarly, these classifiers could provide probabilities of 0.6 that e_1 is a medical kit, 1.0 that e_3 is a book, 1.0 that e_1 is "near" e_3, and 0.66 that e_1 is "near" e_2, resulting in the final likelihoods under an assumption of independence between constraints.

Normalizing these likelihoods through the process described above will produce the following DS-theoretic *Basic Belief Assignment(BBA)* assigning probability masses to each hypothesis in Θ, where Bl and Pl (belief and plausibility) are upper and lower bounds on the expected probability of each hypothesis:

Hypothesis	Mass	Bl	Pl
$\emptyset$	0.0	0.0	0.0
$\{\theta_1\}$	0.2	0.2	0.2
$\{\theta_2\}$	0.3	0.3	0.3
$\{\theta_3\}$	0.5	0.5	0.5
$\{\theta_1, \theta_2\}$	0.0	0.5	0.5
$\{\theta_2, \theta_3\}$	0.0	0.8	0.8
$\{\theta_3, \theta_1\}$	0.0	0.7	0.7
$\{\theta_1, \theta_2, \theta_3\}$	0.0	1.0	1.0

Through the calculations above, we are thus able to produce a DS-theoretic *Frame of Discernment (FoD)* Θ of hypotheses described by the logical conjunctions (i.e., BUSSes) $\{\psi_0 \ldots \psi_n\}$.[5] Remember that each BUSS contains both a parsed utterance form and a set of supplemental semantics, bound using a single candidate variable binding Γ_i. The next component in the DIARC NL Pipeline (i.e., the Pragmatics Component, PRAG) only uses the utterance form, however, and there may be multiple hypotheses in the resulting Frame of Discernment Θ that have the same utterance form but different supplemental semantics.

For instance, in the example used in this section, one candidate medkit (e_1) is actually near two books (e_2 and e_3), and accordingly, two hypotheses are described by BUSSes that have the same utterance form (e.g., $QuestionYN(b, s, grab(s, e_1))$) but different supplemental semantics (e.g., $\{medkit(e_1) \wedge book(e_2) \wedge near(e_1, e_2)\}$ vs $\{medkit(e_1) \wedge book(e_3) \wedge near(e_1, e_3)\}$. We thus cluster these hypotheses into sets $C_0, \ldots, C_n$ such that all hypotheses associated with each set are described by BUSSes that have the same utterance form, an operation linear in the number of referential hypotheses. For example, if we have three singleton hypotheses $\{\theta_1, \theta_2, \theta_3\}$, and ψ_1 and ψ_2 have the same utterance form, $C = \{\{\theta_1, \theta_2\}, \{\theta_3\}\}$.

We can now split our FoD Θ into a set of $|C|$ "binary" FoDs, one for each cluster C_i. Each binary FoD itself has two hypotheses: (1) that the utterance form describing all hypotheses in cluster C_i represents what was communicated, and (2) that it does not. This splitting has no theoretical ramifications, but enables easier integration with PRAG. Moreover, the use of binary FoDs significantly reduces the complexity of inference, precluding the need for DS-theoretic approximate inference algorithms (Bauer 1997) or other efficiency improvements (Polpitiya et al. 2017). Because each cluster is mutually exclusive from all other clusters, each binary

[5] Note, however, that low-probability hypotheses are pruned out during the resolution process, and thus the remaining hypotheses have a higher concentration of mass (and thus, higher belief and plausibility) than they would if this pruning process were not employed. This pruning process is further described by Williams et al. (2016).

FoD can be represented entirely by the *bound utterance structure*:

$$\langle utterance(\psi_i),\ Bl(\{C_{i_0} \dots C_{i_n}\}),\ Pl(\{C_{i_0} \dots C_{i_n}\})\rangle.$$

calculated as

$$\langle utterance(\psi_i),\ \sum_{j=0}^{|C_i|} m(C_{i_j}),\ \sum_{j=0}^{|C_i|} m(C_{i_j})\rangle.$$

In our example, for instance, because ψ_1 and ψ_2 have the same utterance form, $C = \{\{\theta_1, \theta_2\}, \{\theta_3\}\}$. From this, the following set of bound utterance structures will be created:

$$\begin{aligned}
&\{\langle QuestionYN(b, s, can(s, grab(s, o_1))),\\
&\quad Bl(\{\theta_1, \theta_2\}),\ Pl(\{\theta_1, \theta_2\})\rangle,\\
&\langle QuestionYN(b, s, can(s, grab(s, o_4))),\\
&\quad Bl(\{\theta_3\}),\ Pl(\{\theta_3\})\rangle\}\\
&\quad = \{\langle QuestionYN(b, s, can(s, grab(s, o_1))), 0.5, 0.5\rangle\\
&\langle QuestionYN(b, s, can(s, grab(s, o_4))), 0.5, 0.5\rangle\}
\end{aligned}$$

5.1.4 Pragmatic inference

In the previous section, the output of reference resolution was converted into a form which could be sent to our DS-theoretic pragmatic reasoning component (PRAG). Accordingly, after this conversion, we send the resulting set of bound utterances structures to PRAG, which uses context to determine the intentions underlying utterances (Williams et al. 2015), producing a set of intentional structures $\langle I, Bl(I), Pl(I)\rangle$. This operation is of complexity $O(rb)$, where r is the number of pragmatic rules and b is the number of bound utterance structures. If the difference between $Bl(I)$ and $Pl(I)$ is sufficiently large, or if $\frac{Pl(I)+Bl(I)}{2}$ is sufficiently close to 0.5, (assessed using Núñez et al. (2013a)'s uncertainty measure shown in Eq. 3 [and further discussed by Williams et al. (2014)]), intention I is deemed in need of clarification.

$$1 + \frac{Pl(I)}{K} log_2 \frac{Pl(I)}{K} + \frac{1 - Bl(I)}{K} log_2 \frac{1 - Bl(I)}{K} \quad (3)$$
$$where K = 1 + Pl(I) - Bl(I).$$

PRAG then formulates an intention-to-know (*itk*) which of these intentions is correct, denoted $itk(s, or(i_0, i_1, \dots, i_n))$. Note that before integration with POWER, PRAG only handled *pragmatic* uncertainty. Because PRAG now receives a set of candidate utterance forms with potentially different argument bindings, it now also automatically handles *referential* uncertainty.

In the future, it will be important to investigate how these may be even more tightly integrated; if pragmatic analysis is carried out concurrently with the resolution process, it may be possible to determine that resolution can stop early, if, e.g., it is determined that an expression is ambiguous and more options have already been found than should be enumerated by the robot, or if it is determined that regardless of what entity is being referred to, the robot cannot or should not carry out the command in which the entity was referenced. That being said, a tighter integration between reference resolution and pragmatics would require making certain trade-offs. On the one hand, a tighter integration might lead to increased efficiency and accuracy, and may align better with recent psychological models of the interplay between reference and pragmatics (e.g. Huang and Snedeker 2011). On the other hand, a tighter integration between any two architectural component necessarily introduces additional complexity, prevents the use of one component without the other, and accordingly, makes it difficult to separate the performance of the two components.

Before we move on, we would like to point out that because DIARC's reference resolution component handles *open worlds*, instances in which interlocutors refer to previously unknown entities do not automatically generate clarification requests. For example, if the robot is told "Go to the room at the end of the hall" and does not know of a room at the end of the hall, it will not ask for clarification, but will rather hypothesize a new location, and carry on. We do not regard such situations as referentially ambiguous (although it may be valuable to ask for more information about this location). Here, the robot knows what entity is being referred to: a previously unknown room at the end of the hall. Similar logic applies in the case of indefinite noun phrases. If the robot is told "Bring me a medical kit", and it knows of multiple medical kits, perhaps containing different supplies, it may indeed be worthwhile to ask for clarification. However, this command is not ambiguous in and of itself, and as such we do not *automatically* ask for clarification as part of the language understanding process when such forms are used by human teammates.

5.2 Decision to communicate

Currently, any *intention-to-know (itk)* formulated during the previous stages is automatically asserted into the robot's knowledge base, triggering a decision to communicate this intention once it is acceptable for the robot to accept the conversational turn. When this decision is made, the *itk* is passed to the pragmatic *generation* component for processing.

We note, however, that there may be many instances in which the robot may elect not to immediately communicate its own intentions to know, or may elect not to satisfy its interlocutors' intentions to know. There may be many situations in which it would be inopportune for a robot to ask for clarification, e.g., if its interlocutor is busy, already under high workload, or is unable to be able to answer the robots' questions (see, for example, Cai and Mostofi 2016; Rosen-

thal et al. 2012b; Rosenthal and Veloso 2012). In addition, it may be inappropriate to ask for clarification (or to respond to a request for information) for ethical grounds (Williams 2018a; Williams and Jackson 2018). All of these considerations represent interesting and important directions for future work.

5.3 Utterance choice

The robot must now determine a contextually appropriate way to formulate its intention at the *utterance level*, determining the illocutionary force (e.g., statement, question, command) and high-level phrasing (including indirectness strategies) that will be most effective to communicate the intended semantic content.

This is accomplished once again by PRAG, which uses the same set of rules for generation as it uses for inference (Williams et al. 2015). In Experiment One, we observed that if there were more than two options, listing those options was dispreferred over a more general question. Thus, if we are to send a clarification request to PRAG that has semantics of the form $itk(self, or(option_1, ...option_n))$, we first check whether or not n is greater than the acceptable number of candidates to list, i.e., two. If $n = 2$, this intention is sent directly to PRAG. Otherwise, all options are unified into a single predicate whose only bound arguments are those that are identical for all options. For example, if $\{option_1, option_2, option_3\} = \{need(jim, obj_1), need(jim, obj_2), need(jim, obj_3)\}$, these will be unified into $need(jim, ?)$, and the intention $itk(self, need(jim, ?))$ will be sent to PRAG instead.

Using DS-theoretic logical operators, PRAG determines a set of candidate utterance forms, each of which is forward-simulated through pragmatic inference to ensure that the agent does not accidentally communicate anything it does not actually believe to be true as a side effect of communicating its primary illocutionary point. The best candidate utterance is then sent to NLG for surface realization.

While this type of forward simulation has been previously undertaken for other stages of the language generation process (e.g., for referring expression generation (Brick and Scheutz 2007; Tellex et al. 2014; Orita et al. 2015)), we believe this is the first use of such a simulation step during the utterance choice stage.[6] What is more, doing so during this stage goes beyond simply increasing the accuracy of language generation, but also serves as a safeguard against utterances that could not only misalign the mental models shared by a robot and its teammates, but which could negatively affect teammates' perceptions of the robot. That being said, the proposed mechanism only provides such a safeguard against false implications that are explicitly represented on the right-hand side of a robots' pragmatic rules, typically as the interpretations of conventionalized indirect speech acts.[7] Possible false implications not prevented by our approach are discussed in (Williams 2018a, b; Williams and Jackson 2018).

5.4 Content selection

Once an appropriately phrased utterance form is chosen by the pragmatic generation component, it is sent to the *Referring Expression Generation* Component for Content Selection. The job of this component is to select the properties that will be used to describe target referents. For example, consider the utterance form

$$QuestionWH(s, b, or(need(b, grab(s, e_1)), need(b, grab(s, e_2)))).$$

Ultimately, this will be translated to something like "Do you need me to grab $\langle e_1 \rangle$ or $\langle e_2 \rangle$?"[8] It is the job of the Referring Expression Component to choose property sets that will be used in the description of e_1 and e_2, e.g., $\{mug(X) \wedge white(X)\}$, or $\{mug(X) \wedge black(X) \wedge large(X)\}$.

Selecting such property sets presents a challenge for robots operating in realistic human–robot interaction scenarios. Classic Referring Expression Generation (REG) algorithms like the Incremental Algorithm (Dale and Reiter 1995) cannot be straightforwardly applied for a variety of reasons; most crucially, because they do not account for uncertain information. And while there have been some previous approaches to referring expression generation under uncertainty (Zarrieß and Schlangen 2016; Roy 2002; Meo et al. 2014), these are specifically designed for referring to

[6] Some other groups have, since the publication of our original work on this topic (Williams et al. 2015), followed a similar approach, notably in the Rational Speech Act Theory inspired robotics literature (Fried et al. 2017) and in work on "inverse semantics" (Knepper et al. 2015, 2017). See also both prior and posterior work on language understanding from the Rational Speech Act psychological literature (Goodman and Stuhlmüller 2013; Goodman and Frank 2016), as well as critiques of such approaches (Gatt et al. 2013; Qing and Franke 2015).

[7] It is important to note that our pragmatic reasoning system currently is only equipped to handle *conventionalized* Indirect Speech Acts. For a comprehensive handling of ISAs, it will be necessary to integrate this approach with a plan reasoning system (Perrault and Allen 1980; Briggs and Scheutz 2013; Trott and Bergen 2017).

[8] Note here that we have chosen to use rules, in our example as well as in our evaluation, that use the form "Do you need Y or Z" rather than the more indirect and hence more polite "Would you like Y or Z". These two forms trade off between our desiderata. "Do you need Y or Z" (in response to "I need X" better demonstrates intentions, but is less pragmatically appropriate, than "Would you like Y or Z", and vice versa. Although we are able to generate both forms using the presented approach, we chose to use the form "Do you need Y or Z", in part because, while it may be less pragmatically appropriate than "Would you like Y or Z", both forms are significantly more pragmatically appropriate than the use of a direct command.

objects in visual scenes, and as such, cannot generate references to entities that are not currently visible or which (in the case of, e.g., goals, ideas, and utterances), *cannot* be visualized.

To address these challenges, we presented *DIST-PIA* (Williams and Scheutz 2017a), an algorithm inspired by the classic Incremental Algorithm, but tailored to realistic robotics applications by using the same Consultant framework used during reference resolution (Williams 2017a; Williams and Scheutz 2016a) to handle uncertain, heterogeneous, and distributed knowledge. The pseudocode for this algorithm is presented below in Algorithms 1 and 2 with detailed in-line comments. For a thorough explanation and demonstrative example, we direct the interested reader to (Williams and Scheutz 2017a).

Notation

C	A set of *Consultants* $\{c_0, \ldots, c_n\}$
c_m^{Λ}	The set of formulae $\{\lambda_0, \ldots, \lambda_n\}$ advertised by the Consultant $c \in C$ responsible for m.
M	A robot's *world model* of entities $\{e_0 \ldots e_n\}$ found in the domains provided by C.
D	The incrementally built up description, comprised of mappings from entities M to sets of pairs (λ, Γ) of formulae and bindings for those formulae.
D^M	The set of entities $m \in M$ for which sub-descriptions have been created.
d^M	The set of entities $m \in M$ involved in sub-description d.
P	The set of candidate (λ, Γ) pairs under consideration for addition to a sub-description.
Q	The queue of referents which must be described.
X	The incrementally pruned set of distractors

Algorithm 1 *DIST-PIA*(m, C)

$D = \text{new Map()}$ // *The Description*
$Q = \text{new Queue}(m)$ // *The Referent Queue*
while $Q \neq \emptyset$ **do**
  // *Consider the next referent*
  $m\prime = \text{pop(Q)}$
  // *Craft a description d for it*
  $d =$ *DIST-PIA-HELPER*$(m\prime, C)$
  $D = D \cup \{m \rightarrow d\}$
  // *Find all entities used in d*
  for all $m\prime\prime \in d^M \backslash keys(D)$ **do**
    // *And add undescribed entities to the queue*
    $push(Q, m\prime\prime)$
  end for
end while
return D

Algorithm 2 *DIST-PIA-HELPER*(m, C)

$d = \emptyset$ // *The Sub-Description*
$X = M \backslash m$ // *The Distractors*
// *Initialize a set of properties to consider: those advertised by the Consultant c responsible for m*
$P = [\forall \lambda \in c_m^{\Lambda} : (\lambda, \emptyset)]$
// *While there are distractors to eliminate or properties to consider*
while $X \neq \emptyset$ and $P \neq \emptyset$ **do**
  $(\lambda, \Gamma) = pop(P)$
  // *Find all unbound variables in the next property*
  $V = find\_unbound(\lambda, \Gamma)$
  if $|V| > 1$ **then**
    // *If there's more than one, create copies of that property under all possible variable bindings that leaving unbound exactly one variable of the same type as the target referent*
    for all $\Gamma\prime \in cross\_bindings(\lambda, \Gamma, C)$ **do**
      // *And push them onto the property list*
      $push(P, (\lambda, \Gamma\prime))$
    end for
    // *Otherwise, if it is sufficiently probable that the property applies to the target referent...*
  else if $apply(c_m, \lambda, \Gamma \cup (v_0 \rightarrow m)) > \tau_{dph}$ **then**
    // *And it's sufficiently probable that it does* ***not*** *apply to at least one distractor...*
    $\bar{X} = [x \in X \mid apply(c_x, \lambda, \Gamma \cup (v_0 \rightarrow x)) > \tau_{dph}]$
    // *Then bind its free variable to the target referent, and add it to the sub-description...*
    if $\bar{X} \neq \emptyset$ **then**
      // *And remove any eliminated distractors*
      $d = d \cup (\lambda, \Gamma \cup (v_0 \rightarrow m))$
      $X = X \backslash \bar{X}$
    end if
  end if
end while
return d

5.5 Surface realization, and speech synthesis

The final stage of clarification request generation is to send the Bound Utterances with Supplemental Semantics produced by Content Selection to the *Natural Language Generation* Component for Surface Realization, and then to the *Speech Production* Component for Speech Synthesis.

For the first of these two stages, we use the open source SimpleNLG package (Gatt and Reiter 2009) to translate an REG-produced BUSS into a textual form such as "Do you need the white mug or the large black mug?" when there are two referential candidates, or "Which one do you need?" in the case of a larger number of referential candidates. Given the types of sentences we generate in this work, translation from utterance form to text is relatively straightforward using SimpleNLG. Utterance type is used to determine clause type (e.g., an Utterance of type `QuestionYN` is used to generate a Sentence with SimpleNLG type `InterrogativeType.YES_NO`); supplemental semantics are used to add modifiers and prepositional phrases based on predicate arity, which are in turn used by SimpleNLG to create complex noun phrases; finally, "and" and "or" predicates are used to create coordinated phrases with subclauses for each predicate argument.

For the second of these stages, the open source MaryTTS package (Schröder and Trouvain 2003; Schröder et al. 2011) is then used to synthesize this text into an audio form that

is produced by the robot. We note, however, that given the modular nature of the DIARC architecture and the loose coupling between NLG and TTS, this particular TTS Component can easily be substituted out for other TTS Components, if desired [e.g., Festival (Black et al. 1998)].

6 Demonstration

To demonstrate the operation of the presented approach, we present proof-of-concept interactions in two simulated environments. These demonstration highlights the full implementation of all stages of the clarification request generation framework through components of the DIARC architecture. In Sect. 6.1, we present a demonstration in a hypothetical office environment, in which the robot does not have full certainty with respect to the properties of objects found in its environment. This demonstration is conducted entirely within DIARC. In Sect. 6.2, we present a demonstration in a simulated alpine search and rescue environment, in which the robot (an aerial search and rescue drone) *does* have full certainty with respect to the properties of objects found in its environment. This demonstration is conducted in a novel hybrid architecture, where natural language processing and goal-directed reasoning is performed in DIARC, whereas reasoning about and processing knowledge in large scale outdoor missions is performed in the KnowRob system. In order to enable robotic systems to autonomously perform tasks without being controlled with or teleoperated through any human low-level commands, we use the CRAM cognitive robot architecture. CRAM enables robotic systems to ground symbolic descriptions into robots' action-perception domains in order to autonomously execute instructions in a way that is sensitive to the context of the mission.

6.1 Demonstration one

Our first demonstration uses the components of the DIARC architecture shown in Fig. 2, as well as components responsible for the simulation of a Pioneer robot within an office environment, and a set {AGENTS, SPEX, OBJECTS} of POWER Consultants (Williams and Scheutz 2015, 2016a; Williams 2017a) providing information about people, places, and things, respectively.

The interaction begins with the speaker saying to the robot "I need the medkit" in an environment in which the robot knows of two medkits, one red and one white. ASR sends this sentence to NLP, which parses the utterance into the dependency tree: *[rootVB like [ncsubj I] [aux would] [dobj ball [det the]]]*.

From this tree, NLP extracts root semantic content $need(X1, X2)$, with utterance type $Statement$, additional semantic content $\{speaker(X1) \wedge medkit(X2)\}$, and presumed cognitive statuses $\{X1 \rightarrow definite, X2 \rightarrow definite\}$. Using this information, POWER searches for the referents to bind to $X1$ and $X2$; for $X1$, POWER finds a single probable candidate: agt_1, with probability 1.0; for $X2$, two candidates are found: obj_1, with probability of satisfaction 0.82, and obj_2, with probability of satisfaction 0.92.[9] These bindings are then used to create the following bound utterances[10]:

$$\{Stmt(b, s, need(b, obj_1)), Stmt(b, s, need(b, obj_2))\}$$

with corresponding probabilities[11] 0.82 and 0.92, respectively. These are normalized and used to create DS-theoretic bound utterance structures, which are passed to DIALOGUE:

$$\{\langle Stmt(b, s, need(b, obj_1)), 0.471, 0.471\rangle, \langle Stmt(b, s, need(b, obj_2)), 0.529, 0.529\rangle\}$$

PRAG possesses the rule:

$$\langle Stmt(X, Y, need(Z, W)) \Rightarrow goal(Y, bring(Y, W, Z)), 0.9, 0.99\rangle, \quad (4)$$

indicating that the robot is between 90 and 99% confident in the rule[12] because the antecedent of this rule matches the utterance form of each bound utterance structure, uncertain Modus Ponens is applied in both cases, producing the set of intentional structures:

$$\{\langle goal(s, bring(s, obj_1, b)), 0.424, 0.576\rangle, \langle goal(s, bring(s, obj_2, b)), 0.476, 0.524\rangle\}$$

Note that at this point, belief no longer equals plausibility: while the robot may not have encoded any ignorance with respect to what utterance was heard, ignorance encoded with respect to the context and rules the robot uses for pragmatic inference are reflected in the uncertainty intervals of the rules' consequents, thus painting a better picture of how much the robot truly knows about its interlocutor's intentions.

[9] As above, the probabilities of different properties holding for these objects were arbitrarily hand-selected for the sake of a clear and simple demonstration walkthrough. A set of "dummy" Consultants were used that provided these hand-selected probabilities when asked for probability judgments. In practice, these probability judgments can be provided by arbitrary classifiers, such as those commonly used for object recognition (e.g. Redmon et al. 2016), which may often return different levels of confidence for different observed objects.

[10] Here, agt_1 is changed to the agent's name for dialogue processing.

[11] All beliefs and plausibilities in this section are rounded.

[12] The uncertainty intervals associated with different rules were arbitrarily hand-selected for the sake of the demonstration walkthrough. For a discussion of how these intervals might be adapted over time, we direct the interested reader to (Williams et al. 2014).

Nunez' uncertainty rule determines that both of these intentions are highly uncertain. DIALOGUE thus determines its own intention to know which is correct, encoded as the structure:

$$\langle itk(s, or(goal(s, bring(s, obj_1, b)), goal(s, bring(s, obj_2, b)))), 1.0, 1.0\rangle$$

To decide how to communicate this intention, the bound utterance structure is passed through PRAG in reverse (Williams et al. 2015), using a rule of the form:

$$\langle QuestionWH(X, Y, or(Z, W)) \Rightarrow itk(X, or(Z, W)), 0.95, 0.95\rangle, \quad (5)$$

Our approach allows recursive generation, and thus Eq. 5 is chained with Eq. 4 to produce:

$$QuestionWH(s, b, or(need(b, obj_1), need(b, obj_2))).$$

This utterance is then sent to NLG for generation of REs for "bob", "obj_1" and "obj_2", and subsequent realization of the entire expression. This produces the text "Do you need the white medkit or do you need the red medkit?" which is then synthesized and output by the robot.

6.2 Demonstration two

Our second demonstration uses a novel hybrid architecture comprised of (1) the DIARC architecture (Schermerhorn et al. 2006), used for goal-oriented reasoning, natural language understanding, and natural language generation, (2) the open-source ROS-based (Quigley et al. 2009) KnowRob system (Tenorth and Beetz 2009), for domain-specific knowledge representation and reasoning, and (3) the open source ROS-based CRAM architecture (Beetz et al. 2012), for grounding symbolic descriptions into robots' environmental contexts in order to effectively execute navigation commands. The components of this hybrid architecture are shown Fig. 7. Also seen in this diagram are components of the ROS-based Gazebo simulator (Koenig and Howard 2004), which is used for visualizing the simulated alpine search and rescue task context used for this demonstration, as shown in Fig. 4.

In the remainder of this section, we will begin by describing the components of this hybrid architecture that have not yet been introduced [i.e., KnowRob (in Sect. 6.2.1), CRAM (in Sect. 6.2.2), and their integration (in Sect. 6.2.3)]. We will then discuss how KnowRob and CRAM are integrated with DIARC in Sect. 6.2.4. Finally, we will provide a walkthrough of our clarification request generation process as it plays out in this novel hybrid architecture, in Sect. 6.2.5.

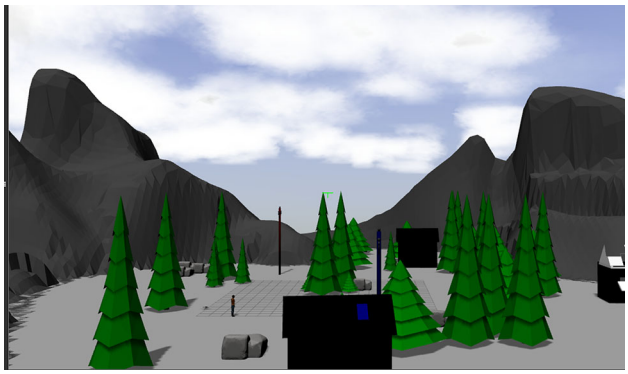

Fig. 4 Visualization of the simulated alpine search and rescue environment in the Gazebo simulator

6.2.1 KnowRob: knowledge representation and reasoning

The KnowRob system provides an efficient reasoning engine equipped with a number of mechanisms important to robot autonomy, e.g., for reasoning about physical objects, actions, and uncertainty (Tenorth and Beetz 2009). To effect such reasoning, KnowRob uses an ontology of structured concepts, properties, and relations. This ontology is implemented in the Web Ontology Language (OWL) (Bechhofer 2009; McGuinness et al. 2004), which formally represents relational knowledge using an XML-like standard. In order to load, store and reason about the knowledge contained in the OWL files, we use SWI-Prolog (Wielemaker 1987; Wielemaker et al. 2012). While KnowRob by default uses an ontology of household concepts, in the context of the outdoor search-and-rescue scenario used for our demonstration, we instead use an extended ontology developed as part of the European SHERPA project (Marconi et al. 2012). This extended ontology includes additional concepts related to alpine search-and-rescue and metric/topological location, some of which are shown in Fig. 5. To provide this additional spatial and domain-specific knowledge, a semantic map was constructed, from which information can be extracted regarding the terrain and features thereof, physical information about objects found within that terrain, and qualitative descriptions of the agents' surroundings. This domain-specific information can be used to augment the knowledge, navigation, task planning, and interaction capabilities of robots operating as human teammates within this task context.

6.2.2 CRAM: cognitive robot abstract machine

To enable robotic systems that are more general, flexible, and reliable than other control systems, we use CRAM (Beetz et al. 2012, 2010): a cognitive robotic architecture that provides a set of mechanisms for reasoning and making decisions about underspecified human instructions.

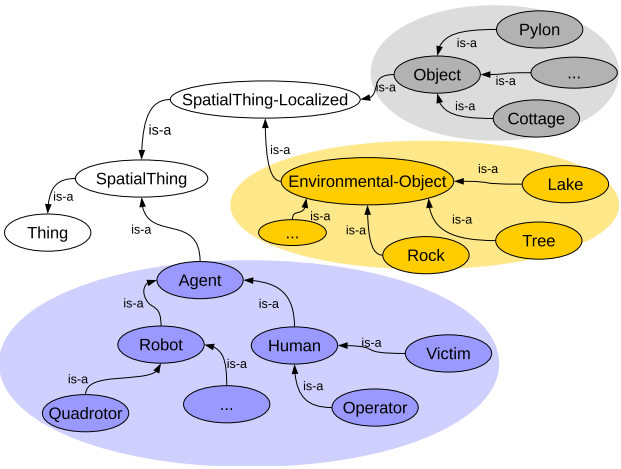

Fig. 5 The KnowRob ontology was extended with additional concepts and classes in order to equip humans and robots with additional knowledge coming from search and rescue applications, including knowledge of terrain (e.g. the concepts of rocks, trees and lakes) and of different agent types (e.g., humans and robots)

Implemented in Common Lisp (Steele 1990), CRAM provides both a plan library and a library of *designators*: symbolic descriptions that specify plan parameters using high-level symbols that, like logical predicates in DIARC, serve as a common currency used throughout disparate parts of the CRAM architecture. These designators can be grounded into robots' perceptions and actions using CRAM's geometric reasoning engine. As an example, a CRAM plan description for the human instruction "Go to the pylon" is shown in Listing 1.

Listing 1 Plan Description of the instruction "Go to the pylon"

```
(an action
  (to go)
  (receiver quadrotor)
  (sender human_operator)
  (destination
    (visible
      (viewpoint human_operator)
      (destination
        (next-to "pylon01")))))
```

This plan description provides an easily executable action designator comprised of two nested location designators equated with a target destination. Those location designators include an object labeled in the robot's semantic map, corresponding with a physical object in the terrain of the alpine search and rescue environment. The designator also contains a set of symbolic constraints that describe the action, the specific robotic teammate assigned to the current task, and conditions regarding the visibility of both the target location and the viewpoint of the robot's human teammate. Finally, this designator also includes the qualitative spatial relation

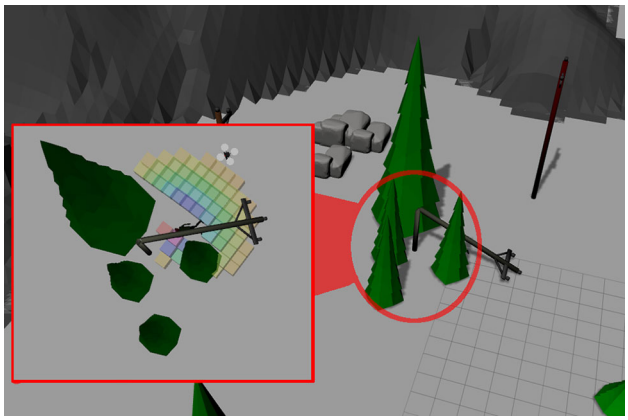

Fig. 6 Satisficing goal locations generated by CRAM. Here, the heatmap in the callout represents the distribution over likely locations for a command to go to the pylon

"next-to" constraint that further restricts the search space of the terrain. These descriptive constraints are obtained from an empirical study in a search and rescue mission (Yazdani et al. 2017) conducted specifically to enable natural tasking of robotic teammates in mixed human–robot teams.

Next, this abstract and qualitative plan description must be translated into a robot motion command to travel to a set of quantitative metric coordinates (Mösenlechner and Beetz 2011). This is achieved by CRAM using a set of Prolog rules which analyze the designator's correctness and generate metric solutions that satisfy symbolic navigational constraints, as depicted in Fig. 6. The robot then selects and travels to one of these satisficing goal locations.

6.2.3 Inter- and intra-architectural communication

Both KnowRob and CRAM are implemented in the open-source Robot Operating System (ROS) (Quigley et al. 2009) middleware. Like the ADE middleware in which DIARC is implemented, ROS supports the generation of robust robot behavior across heterogeneous robotic platforms, and uses a component-based architecture in which a set of "nodes" are run in parallel. While in ADE, inter-component communication is performed using a service-based model, inter-component communication in ROS is performed both through service calls as well as through message passing through a publish-subscribe model. Interfaces to both KnowRob and CRAM are provide using the open source `json_prolog` ROS package, which allows ROS nodes to issue Prolog queries in a standard JSON format (Crockford 2006). Queries issued to KnowRob and CRAM can be used to, respectively, retrieve information from its semantic map and issue navigation commands.

To simulate alpine search and rescue missions, KnowRob and CRAM are integrated with the ROS-integrated Gazebo

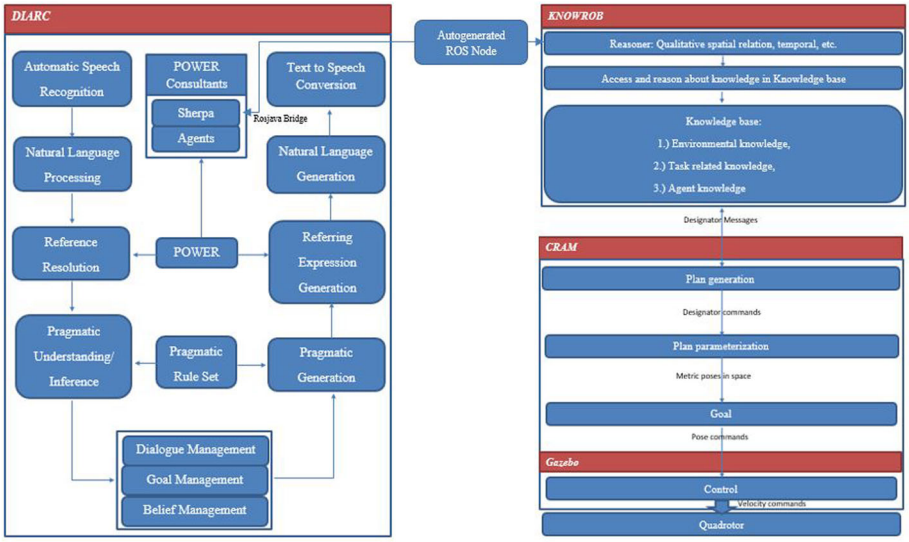

Fig. 7 Integrated architectural diagram: on the left are shown the components of the DIARC architecture used throughout this paper. On the right hand side are the components of KnowRob and CRAM (as well as the Gazebo simulator) introduced in this section. Between them sits the autogenerated ROS Node injected by DIARC into KnowRob+CRAM. Further details can be found in the text

Simulator (Koenig and Howard 2004): a multi-robot simulator suitable for both indoor and outdoor environments, in which we simulate robots' physical activities and robots' perceptions of the objects in their environmental context.

6.2.4 Architectural integration

While DIARC and KnowRob+CRAM share an overlapping set of capabilities, each provides its own unique capabilities that are not found in the other. For example, DIARC provides the natural language processing capabilities previously detailed in this paper, and KnowRob+CRAM provides a unique ontology of search and rescue oriented knowledge, as well as mechanisms for grounding commands to a robot in an alpine search and rescue scenario.

In this section, we will describe how DIARC and KnowRob+CRAM are integrated together so that each architecture may make use of the other's complementary capabilities, as shown in Fig. 7. To integrate the DIARC and KnowRob+CRAM architectures together, we use a strategy we refer to as "component injection", in which the Sherpa Component[13] (a dedicated component of the DIARC Architecture) interacts with an autogenerated ROS node which is injected into the ROS-based KnowRob+CRAM architecture to enable inter-architectural communication. In this section, we will begin by discussing the KnowRob Component. We will then describe how the ROS node it makes use of is created and injected into the KnowRob+CRAM architecture. Finally, we will discuss how we use this connection to allow DIARC and KnowRob+CRAM to leverage each others' capabilities.

[13] This Component is named after the European SHERPA project (Marconi et al. 2012) for which the alpine search and rescue KnowRob ontologies used in this integration were developed.

The Sherpa Component serves as a POWER Consultant to the rest of DIARC (see Sect. 5.1.1). Accordingly, this component advertises the properties that can be handled by KnowRob , provides upon request a list of entities currently known of by KnowRob , and provides probability judgments regarding whether certain properties hold for certain entities, assessed through queries to KnowRob . Many of these capabilities require translating the names of entities between the format used by KnowRob+CRAM and the format used by DIARC. Specifically, KnowRob+CRAM refer to entities by their type, through names such as "tree01" and "rock03". DIARC, in contrast, refers to entities using pairs of the form $< Consultant, id >$ where $Consultant$ is the name of the POWER Consultant responsible for the entity, and id is an integer id assigned by that Consultant to that entity. We will now describe how the Sherpa Component implements each of the capabilities required of POWER Consultants.

First, all POWER Consultants must be able to provide a list of properties that can be assessed in their associated knowledge base. When the Sherpa Component is initialized, it automatically sends a query to KnowRob+CRAM asking for a list of properties and relations that can be assessed during the processes of reference resolution and referring expression generation. Crucially, this means that the Sherpa Component does not need to store any domain-specific knowledge about these properties. Accordingly, when the KnowRob ontology is updated with new search and rescue relevant properties, the Sherpa Component is automatically informed of these new properties at startup.

Second, all POWER Consultants must be able to provide a list of entities currently known of to their associated knowledge base. When POWER asks the Sherpa Component for a list of entities known about by KnowRob+CRAM, the Sherpa Component makes a query for this same list to KnowRob+CRAM. It then updates its mapping between

DIARC-style ID pairs and Sherpa-style entity names, creating new ID pairs for any entities it did not previously know of.

Third, all POWER Consultants must be able to provide probability judgments as to whether particular properties and relations hold for particular entities. When the Sherpa Component receives such a request, it translates it into a JSON Prolog query, which is then sent to KnowRob+CRAM through the architectural bridge we will go on to describe in this section. Because Prolog does not represent the uncertainty of knowledge, KnowRob+CRAM returns to the Sherpa Component a Boolean representing whether or not the queried property appears in (or can be inferred from) KnowRob+CRAM's Prolog knowledge base. This is then translated by the Sherpa Component to a probability value of either 0.0 or 1.0. As such, in this architectural integration, Dempster-Shafer theoretic, point-probability theoretic, and Boolean representations are used seamlessly alongside each other.

Finally, all POWER Consultants must be able to assert new hypothetical representations into their associated knowledge bases. When this is requested of the Sherpa Component, it creates a new DIARC-style ID pair, associates it with a new Sherpa-style entity name beginning with the prefix "hyp" which is unused by KnowRob+CRAM. The Sherpa Component then issues a JSON Prolog request to KnowRob+CRAM to assert the existence of a new entity with this name, along with the properties attributed to it, into its Prolog knowledge base.

In addition to these Consultant capabilities, KnowRob provides search and rescue specific action specifications to DIARC that can be automatically identified and executed by DIARC's Goal Manager, such as commands to navigate an aerial search and rescue robot to a particular location or to command it to take a picture using its onboard camera. When these actions are executed, they trigger associated CRAM commands (through mechanisms we will shortly describe) which determine how to execute that action given the robot's current context.

To allow the Sherpa Component to interact with KnowRob +CRAM, we use an offline utility to automatically generate a ROS Node that can serve as a bridge between DIARC and KnowRob+CRAM [see also (Wilson et al. 2016)]. Specifically, by pointing this offline utility towards the JSON Prolog node of KnowRob+CRAM, a new ROS node is created that can both communicate with DIARC through a rosjava bridge and also communicate with KnowRob+CRAM by passing messages to the JSON Prolog node. While DIARC is fully aware that this node is part of an external ROS-based architecture, from the perspective of KnowRob+CRAM it is simply yet another ROS node publishing and subscribing to messages. KnowRob+CRAM is thus unaware that it is communicating with another robot architecture (i.e., DIARC).

It is interesting to contrast this Component Injection paradigm with the Dual Citizen paradigm introduced in our previous work (Williams et al. 2017). In that previous work, so-called "Dual Citizen" components were true components of both architectures, and accordingly, neither architecture was aware of the Dual-Citizen status of these components, or in fact of the presence of the other architecture at all. In contrast, while the ROS-based KnowRob+CRAM is not aware of the ADE-based DIARC, DIARC is in fact aware that it is interacting with another architecture through the use of this autogenerated ROS node.

6.2.5 Demonstration walkthrough

Now that we have detailed the components of this novel hybrid architecture, we can present a second demonstration of this paper's integrated approach, within a very different task context and evaluation environment. While in the first demonstration, all knowledge was hard-coded into components specifically designed for that demonstration (and accordingly was not grounded into any situated context), in this demonstration, the robot's knowledge is actually grounded into the simulation environment provided by the KnowRob+CRAM architecture.

In the previous demonstration, we used the indirect request "I need the medkit." In this demonstration, we use instead the direct command "Go to the pylon," in an environment containing (among other scenario-relevant objects, locations, and people, as shown in Fig. 4) two pylons: a red pylon, and a blue pylon. To begin, ASR sends this command to NLP to be parsed. While in the previous demonstration, we used the CNC dependency parser to translate the request into a dependency tree from which logical formulae were extracted, in this demonstration, we instead use an in-house Combinatory Categorical Grammar (CCG) (Steedman and Baldridge 2011) parser to incrementally translate text to a logical form. While both parsers may still be used in our system, since performing our initial demonstration we have moved to primarily use this CCG-based parser in order to exploit the incrementality and other benefits it affords, as described by Krause et al. (2013) and Scheutz et al. (2017).

Specifically, NLP parses this command into a bound utterance with type Instruction, with root semantic content $move(self, X0)$, additional semantic content $\{pylon(X0)\}$, and presumed cognitive statuses $\{X0 \rightarrow definite\}$. POWER then uses the Sherpa Component to retrieve a list of currently known entities, and to assess whether each of them has the property $pylon(X0)$ using the JSON Prolog query `check_object_property([entity name],pylon, A)`. Sherpa identifies two candidates, $redpylon01$ and $bluepylon01$ (which the Sherpa Component knows of as $sherpa\_14$ and $sherpa\_23$), each with probability of satisfaction 1.0. These bindings are then used to create the

following bound utterances, each with a corresponding probability of 1.0:

$$\{INSTRUCT(speaker, self, move(self, sherpa\_14)),$$
$$INSTRUCT(speaker, self, move(self, sherpa\_23))\}$$
$$\{INSTRUCT(speaker, self, move(self, sherpa\_14)),$$
$$INSTRUCT(speaker, self, move(self, sherpa\_23))\}.$$

These are normalized and used to create DS-theoretic bound utterances, which are passed to DIALOGUE:

$$\{\langle INSTRUCT(speaker, self, move(self, sherpa\_14)), 0.5, 0.5\rangle,$$
$$\langle INSTRUCT(speaker, self, move(self, sherpa\_23)), 0.5, 0.5\rangle\}$$

the Pragmatics Rule Set contains the rule:

$$\langle INSTRUCT(X, Y, move(Y, Z)) \Rightarrow goal(Y, at(Y, Z)), 0.7, 0.7\rangle. \quad (6)$$

Because the antecedent of this rule matches the utterance form of each bound utterance structure, uncertain Modus Ponens is applied in both cases, producing the set of intentional structures:

$$\{\langle goal(self, at(self, sherpa\_14)), 0.35, 0.65\rangle,$$
$$\langle goal(self, at(self, sherpa\_23)), 0.35, 0.65\rangle\}.$$

Nunez' uncertainty rule determines that both of these intentions are highly uncertain. D thus determines its own intention to know which is correct, encoded as the structure:

$$\langle itk(self, or(goal(self, at(self, sherpa\_14)), goal(self, at(self, sherpa\_23))), 1.0, 1.0)\rangle$$

To decide how to communicate this intention, the bound utterance structure is passed through PRAG in reverse, using a rule of the form

$$\langle QuestionWH(X, Y, or(Z, W)) \Rightarrow itk(X, or(Z, W)), 0.95, 0.95\rangle, \quad (7)$$

Our approach allows recursive generation, and thus Eq. 7 is chained with Eq. 6 to produce:

$$QuestionWH(self, speaker, or(move(self, sherpa\_14), move(self, sherpa\_23))).$$

This utterance is then sent to NLG for generation of REs for *sherpa*_14 and *sherpa*_23. To do so, NLG once again leverages the Sherpa Component, issuing requests to KnowRob+CRAM to assess which properties and relations hold for these entities and their distractors. For each of these two referents, the DIST- POWER algorithm will first determine that the property $pylon(X)$ holds, and rules out all non-pylon distractors. It will then determine that the properties $red(X)$ and $blue(X)$, respectively, hold for these two referents while eliminating all other distractors (i.e., each other). NLG then uses this information to generate the final clarification request, in which *sherpa*_14 is referred to as "the red pylon" and *sherpa*_23 is referred to as "the blue pylon".

7 Human-subjects evaluation

To further evaluate our approach, we conducted an additional human-subject experiment, comprised of (1) a data collection stage, and (2) an evaluation stage. The goal of this evaluation was to compare clarification requests authored by our integrated approach with those authored by humans, using human preference judgments as an evaluation metric. While we would not necessarily expect the quality of requests authored by our approach to *surpass* that of requests generated by humans, we would hope and expect that they would approach reasonably close.

7.1 Data collection

We first created a tabletop scene containing twelve objects: four different colored waterbottles, four different colored markers, and four different colored mugs (Fig. 8). This type of tabletop environment is typical of the challenging environment in which natural language dialogue in general (Kruijff et al. 2006a; Scalise et al. 2018; Scheutz et al. 2014) and clarification dialogues specifically (Wyatt 2005) have typically been studied in robotics. Not only is clutter a pervasive and challenging aspect of natural environments (Berenson and Srinivasa 2008), but it produces an environment in

Fig. 8 Tabletop environment used in experiment two

which speakers may not be aware of—or hold in working memory—all distractors which need to be disambiguated in their referring expressions. This has similarly made these types of environments particularly valued in the fields of natural language generation and psycholinguistics (Koolen et al. 2016).

For each object type, we took photographs of the scene in which zero, one, or two of that object type were taken away. This produced nine tabletop scenes, three of which contained identical object arrangements (i.e., those scenes in which no objects were removed). In our data collection experiment, each participant was shown one of these nine images at random, with a caption describing the participant's task, followed by a text box. For example, for the image in which three of the four waterbottles was shown, the following caption was used:

"You have been told 'I need the bottle!' and would like to fulfill the speaker's request. However, as you can see, there are three bottles on the table: a silver bottle, a green bottle, and a blue bottle. Please type a sentence you would use to ask the speaker for clarification, so that you will know what bottle to pick up."

Similar captions were used for the other images. Once the participant entered text into the text box, they were free to click to the next page, and end the experiment. It is important to note that while these captions likely primed participants to use particular features (e.g., object type and color), this is unimportant for our experiment: as we will go on to describe, those features of participants' sentences were not used in the subsequent stage of our experiment. Because we were only interested in—and thus only made use of the sentence content regarding—how the clarification requests themselves were phrased, this priming has no impact on the ecological validity of our experiment.

Participants were recruited (53 Male, 39 Female) using Amazon Mechanical Turk. All participants were American, ranged in age from 20 to 77 (M = 33.15, SD = 8.94), and were paid $0.30 to participate. Only high-reputation participants were used to guard against potential participation from automated "bots". As a total of 92 participants were recruited, an average of 30.7 utterances were collected for each grouping of scenes that had the same number of objects removed. Vocabulary diversity statistics for these utterances are reported in Table 1 (Carroll 1964; Richards 1987). Here, Vocabulary indicates the number of words collected; Types indicates the number of unique words collected; and Type/Token ratio ($\frac{Types}{Vocabulary}$) and Diversity ($\frac{Types}{\sqrt{2*Vocabulary}}$) provide competing metrics of the size of vocabulary used by speakers. These measures may help the reader better gauge the relative complexity of our experimental setup. All utterances collected in this stage were standardized with respect to noun phrasing. For example, "Do you want me to pick up the silver bottle or the blue bottle?" was reduced to "Do _ want _ to pick up _ or _?" All utterances within each cluster were grouped by identical phrasing, and the three most common phrasings for each cluster were selected (four in the case of a tie). The REG algorithm described above and presented in (Williams and Scheutz 2017a) was then used to generate noun phrases to fill into the previously created gaps, thus creating three to four utterances for each image.

Table 1 Vocabulary statistics for utterances collected in Experiment Two, Part One

Vocabulary	Types	Type/token ratio	Diversity
596	64	.107	1.85

Next, an additional utterance was generated for each image using the approach presented in this paper: for each image, knowledge of the objects in the image was provided to the robot architecture, and the utterance "I need the [name of object type]" was said to a robot running the architecture. Because the architecture also used the REG algorithm described above and presented in (Williams and Scheutz 2017a), the utterances generated by our robot architecture had the same noun-level phrasings as all other utterances, but a different utterance-level phrasing. Thus, this stage produced a set of thirty-nine utterances with unique utterance level phrasings but identical noun level phrasings. The thirteen utterance forms (before REs were filled in) are shown in Table 2, Column 3.

7.2 Evaluation

In this stage, a new set of participants were recruited (94 Male, 88 Female) using Amazon Mechanical Turk. These participants were American, ranged in age from 18 to 74 (M = 34.55, SD = 11.16), and were paid $0.30 to participate. Only high-reputation participants were used to guard against potential participation from automated "bots".

Each of these new participants was shown one of the nine tabletop scenes created in the first stage, along with a caption such as: "Your friend Alex says to you, 'I need the bottle!' Which of the following sentences would be best to say to Alex, so that you will know which bottle to give her?"

Each participant was then presented with the four to five utterances associated with the presented image, in the form of buttons. Clicking on one of the utterances moved the participant to the next page, and ended the experiment. As a total of 182 participants were recruited, an average of 20.22 data points were collected for each scene.

As previously mentioned, while we would not necessarily expect the quality of requests authored by our approach to

Table 2 Utterance forms generated in Experiment Two, Part One, and chosen between in Experiment Two, Part Two

Generator	#	Utterance generated in Part One	Result (%)
Robot	2	Do you need __ or do you need __?	9.4
Human	2	Do you need __ or __?	**45.3**
Human	2	What color __ do you need?	22.6
Human	2	What color __ do you want?	22.6
Robot	3	Which one do you need?	23.7
Human	3	Which color __ do you need?	**33.9**
Human	3	Which color __?	23.7
Human	3	Which color __ would you like?	18.6
Robot	4	Which one do you need?	20.0
Human	4	What color __ do you need?	**24.3**
Human	4	Which color __ would you like?	22.9
Human	4	Which color __?	21.4
Human	4	What color is the __?	11.4

The largest percentage for each scene is depicted in bold
Col. 1 indicates whether each utterance form was generated by the presented approach or by a human in Part One. *Col. 2* indicates how many suitable referents existed in the scene for which each utterance was generated. *Col. 3* indicates the generated utterance form, generalized across noun phrases. In Part Two, blanks were filled with generated REs. For example, in scenes with initial utterance "I need the bottle", gaps in the first two rows were filled with "the green bottle" and "the silver bottle", and remaining gaps were filled with "bottle". *Col. 4* indicates the percentage of participants in Part Two who chose that utterance form as the best to use to ask for clarification

surpass that of requests generated by humans, we would hope and expect that they would approach reasonably close. And in fact, Robot-generated requests were chosen only slightly less frequently than were human-generated requests: overall, robot-generated requests were chosen 18.13% of the time, whereas each form of human-generated request was chosen, on average, 24.67% of the time. Overall, this is a positive result as it suggests that the algorithm overall did not generate requests that were much worse than the requests that humans used most frequently. A request-by-request breakdown of participants' choices is shown in Table 2, Column 4.

This table also indicates, however, a more complex story, in the case where there were only two referential candidates. As shown in the first section of Table 2, in this case our robot-generated requests were chosen significantly less frequently than were human-generated requests, but were nearly identical to the top performing human-generated requests. The robot-generated requests were simply more verbose, as they used a conjunction at the clause level rather than the noun-phrase level. We thus subsequently adapted our algorithm to ensure that this type of elision was automatically performed, the result being that our algorithm now produces exactly the form that was most preferred by our participants. Without running a replication experiment with the modified algorithm, however, we cannot make new claims as to the performance of our approach nor re-calculate our statistics, as this would implicitly use a test set of compromised validity.

7.3 Discussion

In Experiment One, we observed that participants dispreferred clarification requests that were insensitive to pragmatic factors, did not indicate understanding of an interlocutor's goals or intentions, listed more than two options, or did not list both options when there were only two likely candidates. These observations were confirmed in Experiment two, part two. The most commonly chosen clarification requests were nearly identical to the clarification requests generated by our robot architecture. But in neither the two-, three-, or four-option utterance groupings were our chosen clarification requests *exactly* identical to the most commonly chosen clarification requests, and in fact differed from those requests in small but important ways.

As previously mentioned, when there was referential ambiguity between only two candidate referents, participants in Experiment Two Part Two preferred clarification requests that listed all options. However, the specific phrasing used by our robot architecture was simply too verbose, as it failed to identify structural similarities and distribute appropriately. As previously mentioned, this fault has since been rectified. Future work will be needed to determine the distribution of selections that would be seen if the overly verbose (originally robot-generated) RE were not presented. We would expect, however, that the most common human-generated (and now, robot-generated) RE to be chosen between 45.3 and 54.5% of the time, putting the robot's performance on par with human performance.

A greater difference is observed when more than two options present themselves. It is striking to observe that all commonly-used human-generated utterances in these cases do not explicitly ask for disambiguation between bottles, but rather ask for information regarding a specific property that could be used to disambiguate between bottles. This suggests that in these cases, it may be advantageous to combine techniques from approaches such as that presented in this paper and the information-theoretic approaches seen in previous work (Deits et al. 2013; Hemachandra et al. 2014; Purver 2004).

It is also important to note, however, that in all three cases a significant percentage of participants did choose the less popular choices. When four options were presented, for example, "Which color__ would you like" was chosen by less than two percent fewer participants than was the most popular "What color __ do you need?". This suggests that it may be valuable in future work to develop models of human interlocutors that model this type of individual difference.

While at first glance the difference between the alternate strategies may seem arbitrary, we suspect that they in fact represent different strategies that are either explicitly used, or which arise from differential weightings of pragmatic principles. Utterances such as "Which color __ *do you need*" may be used due to subconscious *lexical entrainment* or conscious *refashioning* in which speakers use the same phrasing as that used by their interlocutors (Clark and Schaefer 1989; Brennan et al. 2010; Yoon and Brown-Schmidt 2013). Utterances such as "Which color __ *would you like*" and "Which color __ *do you want*" may be used if the pragmatic value of a refashioned sentence is weighted lower than that of a more *conventionally indirect* utterance form (Searle 1975). And utterances such as "Which color __" may be used due to the interaction of either aforementioned pragmatic strategy with Grice (1970)'s Third Maxim of Manner: "Be brief (avoid unnecessary prolixity)".

Before concluding, let us discuss a few limitations of the evaluation presented in this section. First, our instructions may have primed participants to prefer utterances of the form "Do you need" through lexical entrainment. As such, while our evaluation effectively demonstrated human preference for the utterances generated by our approach, it's unclear whether this is due to communication of intentions or due to lexical entrainment. It will be valuable to further evaluate this distinction in future work. More generally, it would be valuable to perform an ablation study to more deeply investigate the benefit gleaned from each of our three initial desiderata, e.g. by increasing the threshold for generalization, by bypassing the pragmatic inference module, and/or by varying the pragmatic rules used during pragmatic generation.

Second, as with our first experiment, there are limitations with respect to experimental setting: in crowdsourced experiments, it is not possible to control participants' experimental setting, and we did not control for web browser, operating system, or other factors that may have impacted viewing experience.

Third, while we only used high reputation participants, we did not use attention checks[14] or Captchas (Von Ahn et al. 2003), which may have been warranted as even further caution against "bots" (Schenk and Guittard 2009). While this was not a concern for the first portion of this experiment, as no participants entered infelicitous responses to our questions, it may have been a concern for our second experiment.

Finally, in the future, it would also be valuable to evaluate our approach (and future extensions thereof) not only subjectively, but objectively as well. While in this work our evaluative focus was on alignment of clarification requests with human preferences, an inherently subjective metric, it will also be important to perform objective evaluation. The NLG community has recently, through community efforts such as the GIVE challenge (Byron et al. 2009; Koller et al. 2010), demonstrated the importance of task-based evaluations where language generation is evaluated with respect to task performance: an approach to evaluation that is objective while avoiding the observed flaws of previous methods such as BLEU scores (Papineni et al. 2002). In our own previous work, we have developed novel task-based evaluation frameworks for evaluating Referring Expression Generation algorithms (Williams and Scheutz 2017a). In future work, it would be a natural next step to investigate how such a framework could be appropriately modified to apply to evaluate the full clarification request generation framework as well.

8 Conclusion

We have presented an integrated approach to clarification request generation for HRI contexts. Our initial experiment replicated and refined the recommendations of previous studies of human–robot dialogue, suggesting that for human–robot interaction contexts, it may be important for robots' clarification requests to be pragmatically appropriate, demonstrate intention understanding, and list options for disambiguation so long as there are a small number of options. In this work, we demonstrated how our integrated approach was able to fulfill these desiderata in two scenarios, including a simulated alpine search and rescue environment enabled through a hybrid architectural approach. In addition, we showed how our approach can be used in architectures where information about referents is uncertain and distributed across multiple heterogeneous knowledge bases, as is often the case in cognitive robot architectures. But

[14] Although, see recent discussion of the shortcomings of such checks (Hauser and Schwarz 2015), especially in crowdsourced experiments (Curran 2016).

most importantly, the primary finding of this paper is that a language-enabled robot's pragmatic reasoning component can track and address *referential* ambiguity when integrated with probabilistic reference resolution and referring expression generation components: a useful finding for designers of language-enabled robot architectures intended for use in HRI domains.

Our findings suggest several directions for future work. First, research is needed on using information-theoretic mechanisms to adapt (Tellex et al. 2013)—and local context (Rosenthal et al. 2012a)—to frame clarification requests generated by pragmatic reasoning components. Second, research is needed to develop speaker-specific models that can predict precisely what type of clarification request they would most likely prefer, based on their inferred weighting of pragmatic principles. Third, future work should also further examine methods by which components using different frameworks for representing uncertainty can be optimally integrated. Fourth, it would be valuable to evaluate the integrated system presented in this paper using task-based and ablation-based evaluations that address the shortcoming of our current evaluations. Finally, a tighter integration between pragmatic reasoning and reference resolution can be achieved. In previous work, we have shown how our pragmatic reasoning component can use contextual knowledge to abduce the most appropriate way to phrase an utterance; but this contextual knowledge is assumed to be stored in a robot's centralized belief and dialogue components. In future work, this should be extended to allow this knowledge to be appropriately distributed across the robot's heterogeneous knowledge bases, as is its other knowledge.

Acknowledgements This work was in part funded by Grant N00014-14-1-0149 from the US Office of Naval Research. The research of Michael Beetz is partly funded by the German science foundation DFG in the context of the collaborative research centre EASE (Everyday Activity Science and Engineering).

References

Bauer, M. (1997). Approximation algorithms and decision making in the Dempster-Shafer theory of evidence—An empirical study. *International Journal of Approximate Reasoning*, *17*(2–3), 217–237.

Bechhofer, S. (2009). Owl: Web ontology language. In *Encyclopedia of database systems* (pp. 2008–2009). New York: Springer.

Beetz, M., Mösenlechner, L., & Tenorth, M. (2010). CRAM—A cognitive robot abstract machine for everyday manipulation in human environments. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Taipei, Taiwan, pp. 1012–1017.

Beetz, M., Mösenlechner, L., Tenorth, M., & Rühr, T. (2012). Cram—A cognitive robot abstract machine. In *5th International conference on cognitive systems (CogSys 2012)*.

Benotti, L., & Blackburn, P. (2017). Modeling the clarification potential of instructions: Predicting clarification requests and other reactions. *Computer Speech & Language*, *45*, 536–551.

Berenson, D., & Srinivasa, S. S. (2008). Grasp synthesis in cluttered environments for dexterous hands. In *Proceedings of the 8th IEEE-RAS international conference on humanoid robots (HUMANOIDS)*, pp. 189–196.

Black, A., Taylor, P., Caley, R., & Clark, R. (1998). *The festival speech synthesis system. Technical report*. Edinburgh: University of Edinburgh.

Brennan, S. E., Galati, A., & Kuhlen, A. K. (2010). Two minds, one dialog: Coordinating speaking and understanding. *Psychology of Learning and Motivation*, *53*, 301–344.

Brenner, M., & Kruijff-Korbayová, I. (2008). A continual multiagent planning approach to situated dialogue. In *Proceedings of the 12th workshop on the semantics and pragmatics of dialogue (Semdial)*, London, UK.

Brick, T., & Scheutz, M. (2007). Incremental natural language processing for HRI. In *Proceeding of the ACM/IEEE international conference on human-robot interaction (HRI)* pp. 263–270.

Briggs, G., & Scheutz, M. (2013). A hybrid architectural approach to understanding and appropriately generating indirect speech acts. In *Proceedings of the twenty-seventh AAAI conference on artificial intelligence (AAAI)*.

Brown, P. (1987). *Politeness: Some universals in language usage* (Vol. 4). Cambridge: Cambridge University Press.

Byron, D., Koller, A., Striegnitz, K., Cassell, J., Dale, R., Moore, J., & Oberlander, J. (2009). Report on the first NLG challenge on generating instructions in virtual environments (GIVE). In *Proceedings of the twelfth European workshop on natural language generation (ENLG), Association for Computational Linguistics*, pp. 165–173.

Cai, H., & Mostofi, Y. (2016). Asking for help with the right question by predicting human visual performance. In *Proceedings of robotics: Science and systems*.

Carrillo, F. M., & Topp, E. A. (2016). Interaction and task patterns in symbiotic, mixed-initiative human-robot interaction. In *AAAI workshop: Symbiotic cognitive systems.*

Carroll, J. B. (1964). *Language and thought. Foundations of modern psychology*. Englewood Cliffs, NJ: Prentice-Hall.

Clark, H. H. (1996). *Using language*. Cambridge: Cambridge University Press.

Clark, H. H., & Schaefer, E. F. (1989). Contributing to discourse. *Cognitive Science*, *13*(2), 259–294.

Crockford, D. (2006). The application/json media type for javascript object notation (json).

Curran, P. G. (2016). Methods for the detection of carelessly invalid responses in survey data. *Journal of Experimental Social Psychology*, *66*, 4–19.

Dale, R., & Reiter, E. (1995). Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, *19*(2), 233–263.

Deits, R., Tellex, S., Kollar, T., & Roy, N. (2013). Clarifying commands with information-theoretic human-robot dialog. *Journal of Human-Robot Interaction (JHRI)*, *2*(2), 58–79.

Dempster, A. P. (2008). The Dempster-Shafer calculus for statisticians. *International Journal of approximate reasoning*, *48*(2), 365–377.

Dzifcak, J., Scheutz, M., Baral, C., & Schermerhorn, P. (2009). What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution. In *Proceedings of the international conference on robotics and automation (ICRA)*, Kobe, Japan.

Fagin, R., & Halpern, J. Y. (1991). A new approach to updating beliefs. In *Uncertainty in artificial intelligence* (pp. 347–374). New York: Elsevier Science Publishers.

Fried, D., Andreas, J., & Klein, D. (2017). Unified pragmatic models for generating and following instructions. arXiv:1711.04987.

Gabsdil, M. (2003). Clarification in spoken dialogue systems. In *Proceedings of the 2003 AAAI spring symposium. Workshop on natural language generation in spoken and written dialogue* (pp. 28–35).

Garoufi, K., & Koller, A. (2014). Generation of effective referring expressions in situated context. *Language, Cognition and Neuroscience*, *29*(8), 986–1001.

Gatt, A., & Reiter, E. (2009). Simplenlg: A realisation engine for practical applications. In *Proceedings of the 12th European workshop on natural language generation, association for computational linguistics* (pp. 90–93).

Gatt, A., van Gompel, R. P., van Deemter, K., & Kramer, E. (2013). Are we Bayesian referring expression generators. In *Proceedings of the thirty-fifth annual meeting of the cognitive science society.*

Ginzburg, J. (2009). The interactive stance: Meaning for conversation (forthcoming in 2009). Studies in Computational Linguistics.

Goodman, N. D., & Frank, M. C. (2016). Pragmatic language interpretation as probabilistic inference. *Trends in Cognitive Sciences*, *20*(11), 818–829.

Goodman, N. D., & Stuhlmüller, A. (2013). Knowledge and implicature: Modeling language understanding as social cognition. *Topics in cognitive science*, *5*(1), 173–184.

Grice, H. P. (1970). Logic and conversation. *Syntax and Semantics*, *3*, 41–58.

Gundel, J. K., Hedberg, N., & Zacharski, R. (1993). Cognitive status and the form of referring expressions in discourse. *Language*, *69*(2), 274–307.

Hauser, D. J., & Schwarz, N. (2015). It's a trap! instructional manipulation checks prompt systematic thinking on "tricky" tasks. *Sage Open*, *5*(2),

Heendeni, J. N., Premaratne, K., Murthi, M., Uscinski, J., & Scheutz, M. (2016). A generalization of Bayesian inference in the Dempster-Shafer belief theoretic framework. In *Proceedings of the international conference on information fusion*.

Hemachandra, S., Walter, M. R., & Teller, S. (2014). Information theoretic question asking to improve spatial semantic representations. In *Proceedings of the AAAI fall symposium series.*

Huang, Y. T., & Snedeker, J. (2011). Logic and conversation revisited: Evidence for a division between semantic and pragmatic content in real-time language comprehension. *Language and Cognitive Processes*, *26*(8), 1161–1172.

Knepper, R. A., Tellex, S., Li, A., Roy, N., & Rus, D. (2015). Recovering from failure by asking for help. *Autonomous Robots*, *39*(3), 347–362.

Knepper, R. A., Mavrogiannis, C. I., Proft, J., & Liang, C. (2017). Implicit communication in a joint action. In *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction (HRI)* (pp. 283–292). ACM.

Koenig, N., & Howard, A. (2004). Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (Vol. 3, pp. 2149–2154).

Kollar, T., Tellex, S., Walter, M., Huang, A., Bachrach, A., Hemachandra, S., et al. (2017). Generalized grounding graphs: A probabilistic framework for understanding grounded commands. arXiv:1712.01097.

Koller, A., Striegnitz, K., Gargett, A., Byron, D., Cassell, J., Dale, R., Moore, J., & Oberlander, J. (2010). Report on the second nlg challenge on generating instructions in virtual environments (GIVE-2). In *Proceedings of the sixth international natural language generation conference (INLG), association for computational linguistics* (pp. 243–250).

Koolen, R., Krahmer, E., & Swerts, M. (2016). How distractor objects trigger referential overspecification: Testing the effects of visual clutter and distractor distance. *Cognitive Science*, *40*(7), 1617–1647.

Krause, E., Cantrell, R., Potapova, E., Zillich, M., & Scheutz, M. (2013). Incrementally biasing visual search using natural language input. In *Proceedings of the 12th international conference on autonomous agents and multi-agent systems (AAMAS)* (pp. 31–38).

Kruijff, G. J. M., Kelleher, J. D., & Hawes, N. (2006a). Information fusion for visual reference resolution in dynamic situated dialogue. In *Perception and interactive technologies*. New York: Springer.

Kruijff, G. J. M., Zender, H., Jensfelt, P., & Christensen, H.I. (2006b). Clarification dialogues in human-augmented mapping. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-Robot Interaction (HRI)* (pp. 282–289).

Kruijff, G. J. M., Brenner, M., & Hawes, N. (2008). Continual planning for cross-modal situated clarification in human-robot interaction. In *Proceedings of the seventeenth IEEE international symposium on robot and human interactive communication (RO-MAN)* (pp. 592–597).

Lemaignan, S., Warnier, M., Sisbot, E. A., Clodic, A., & Alami, R. (2017). Artificial cognition for social human-robot interaction: An implementation. *Artificial Intelligence*, *247*, 45–69.

Marconi, L., Melchiorri, C., Beetz, M., Pangercic, D., Siegwart, R., Leutenegger, S., Carloni, R., Stramigioli, S., Bruyninckx, H., Doherty, P., et al. (2012). The sherpa project: Smart collaboration between humans and ground-aerial robots for improving rescuing activities in alpine environments. In *Proceedings of the IEEE international symposium on safety, security, and rescue robotics (SSRR)* (pp. 1–4)

Marge, M., & Rudnicky, A. I. (2015). Miscommunication recovery in physically situated dialogue. In *Proceedings of the sixteenth annual meeting of the special interest group on discourse and dialogue (SIGDIAL)* (pp. 22–49).

Matarić, M. J. (2002). Situated robotics. In L. Nadel (Ed.), *Encyclopedia of cognitive science*. London: Nature Publishers Group, Macmillan Reference Ltd.

Matuszek, C., Herbst, E., Zettlemoyer, L., & Fox, D. (2012). Learning to parse natural language commands to a robot control system. In *Proceedings of the thirteenth international symposium on experimental robotics (ISER)* (pp. 403–415).

Maurtua, I., Fernandez, I., Kildal, J., Susperregi, L., Tellaeche, A., & Ibarguren, A. (2016). Enhancing safe human-robot collaboration through natural multimodal communication. In: *Proceedings of the 21st IEEE international conference on emerging technologies and factory automation (ETFA), IEEE* (pp. 1–8).

Mavridis, N. (2015). A review of verbal and non-verbal human-robot interactive communication. *Robotics and Autonomous Systems*, *63*, 22–35.

McGuinness, D. L., Van Harmelen, F., et al. (2004). Owl web ontology language overview. *W3C Recommendation 10*(10):2004.

Meo, T., McMahan, B., & Stone, M. (2014). Generating and resolving vague color references. In *Proceedings of the eighteenth SEMDIAL workshop on the semantics and pragmatics of dialogue (DialWatt)* (pp. 107–115).

Mösenlechner, L., & Beetz, M. (2011). Parameterizing actions to have the appropriate effects. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, San Francisco, CA, USA.

Mutlu, B., & Forlizzi, J. (2008). Robots in organizations: the role of workflow, social, and environmental factors in human-robot interaction. In *Proceedings of the 3rd ACM/IEEE international conference on human-robot interaction (HRI)* (pp. 287–294).

Núñez, R. C., Dabarera, R., Scheutz, M., Briggs, G., Bueno, O., Premaratne, K., & Murthi, M. N. (2013a). DS-based uncertain implication rules for inference and fusion applications. In *Proceedings of the sixteenth international conference on information fusion (FUSION)* (pp. 1934–1941).

Núñez, R. C., Scheutz, M., Premaratne, K., & Murthi, M. N. (2013b). Modeling uncertainty in first-order logic: A Dempster-Shafer theoretic approach. In *Proceedings of the eighth international symposium on imprecise probability: Theories and applications*.

Orita, N., Vornov, E., Feldman, N., & Daumé III, H. (2015). Why discourse affects speakers' choice of referring expressions. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing* (Vol. 1, pp. 1639–1649).

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics, Association for Computational Linguistics* (pp. 311–318).

Perrault, C. R., & Allen, J. F. (1980). A plan-based analysis of indirect speech acts. *Computational Linguistics*, *6*(3–4), 167–182.

Polpitiya, L. G., Premaratne, K., Murthi, M. N., & Sarkar, D. (2017). Efficient computation of belief theoretic conditionals. In *Proceedings of the tenth international symposium on imprecise probability: Theories and applications* (pp. 265–276).

Purver, M. (2004). Clarie: The clarification engine. In *Proceedings of the eighth SEMDIAL workshop on the semantics and pragmatics of dialogue (CATALOG)* (pp. 77–84).

Purver, M., Ginzburg, J., & Healey, P. (2003). On the means for clarification in dialogue. In *Current and new directions in discourse and dialogue* (pp. 235–255). New York: Springer.

Qing, C., & Franke, M. (2015). Variations on a Bayesian theme: Comparing Bayesian models of referential reasoning. In *Bayesian natural language semantics and pragmatics* (pp. 201–220). New York: Springer

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T. B., Leibs, J., Wheeler, R., & Ng, A. Y. (2009). ROS: an open-source robot operating system. In *ICRA workshop on open source software*.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788).

Reiter, E., Dale, R., & Feng, Z. (2000). *Building natural language generation systems*. Cambridge: MIT Press.

Richards, B. (1987). Type/token ratios: What do they really tell us? *Journal of Child Language*, *14*(02), 201–209.

Rodríguez, K. J., & Schlangen, D. (2004). Form, intonation and function of clarification requests in German task-oriented spoken dialogues. In *Proceedings of the 8th workshop on the semantics and pragmatics of dialogue (SemDial)*.

Rosenthal, S., & Veloso, M. (2012). Mobile robot planning to seek help with spatially-situated tasks. In: *Proceedings of the AAAI conference on artificial intelligence (AAAI)*.

Rosenthal, S., Veloso, M., & Dey, A. K. (2012a). Acquiring accurate human responses to robots' questions. *International Journal of Social Robotics*, *4*(2), 117–129.

Rosenthal, S., Veloso, M., & Dey, A. K. (2012b). Is someone in this office available to help me? Proactively seeking help from spatially-situated humans. *Journal of Intelligent and Robotic Systems*, *66*, 205–221.

Roy, D. K. (2002). Learning visually grounded words and syntax for a scene description task. *Computer Speech and Language*, *16*(3–4), 353–385.

Scalise, R., Li, S., Admoni, H., Rosenthal, S., & Srinivasa, S. S. (2018). Natural language instructions for human-robot collaborative manipulation. *The International Journal of Robotics Research*, *37*(6), 558–565.

Schegloff, E. A. (1987). Some sources of misunderstanding in talk-in-interaction. *Linguistics*, *25*(1), 201–218.

Schenk, E., & Guittard, C. (2009). Crowdsourcing: What can be outsourced to the crowd, and why. In *Workshop on open source innovation, Strasbourg, France, Vol. 72*.

Schermerhorn, P. W., Kramer, J. F., Middendorff, C., & Scheutz, M. (2006). DIARC: A testbed for natural human-robot interaction. In *Proceedings of the twentieth AAAI conference on artificial intelligence (AAAI)* (pp. 1972–1973).

Scheutz, M., Krause, E., & Sadeghi, S. (2014). An embodied real-time model of language-guided incremental visual search. In:*Proceedings of the thirty-sixth annual meeting of the cognitive science society*.

Scheutz, M., Krause, E., Oosterveld, B., Frasca, T., & Platt, R. (2017). Spoken instruction-based one-shot object and action learning in a cognitive robotic architecture. In *Proceedings of the 16th conference on autonomous agents and multiagent systems (AAMAS)* (pp. 1378–1386).

Schröder, M., & Trouvain, J. (2003). The german text-to-speech synthesis system MARY: A tool for research, development and teaching. *International Journal of Speech Technology*, *6*(4), 365–377.

Schröder, M., Charfuelan, M., Pammi, S., & Steiner, I. (2011). Open source voice creation toolkit for the MARY TTS platform. In *Twelfth annual conference of the international speech communication association.*

Searle, J. R. (1975). Indirect speech acts. *Syntax and Semantics*, *3*, 59–82.

Shafer, G. (1976). *A mathematical theory of evidence*. Princeton: Princeton University Press.

Steedman, M., & Baldridge, J. (2011). *Combinatory categorial grammar. Non-transformational syntax: Formal and explicit models of grammar* (pp. 181–224).

Steele, G. (1990). *Common LISP: The language*. New York: Elsevier.

Stirling, A. (2010). Keep it complex. *Nature*, *468*(7327), 1029–1031.

Stoyanchev, S., Liu, A., & Hirschberg, J. (2013). Modelling human clarification strategies. In *Proceedings of the 14th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)* (pp. 137–141).

Talamadupula, K., Kambhampati, S., Schermerhorn, P., Benton, J., & Scheutz, M. (2011). Planning for human-robot teaming. In *Proceedings of the ICAPS workshop on scheduling and planning applications (SPARK), Vol. 67*.

Tang, Y., Hang, C. W., Parsons, S., & Singh, M. P. (2012). Towards argumentation with symbolic Dempster-Shafer evidence. In *Proceedings of the second international conference on computational models of argument (COMMA)* (pp. 462–469).

Tellex, S., Kollar, T., Dickerson, S., Walter, M. R., Banerjee, A. G., Teller, S., et al. (2011). Approaching the symbol grounding problem with probabilistic graphical models. *AI Magazine*, *32*(4), 64–76.

Tellex, S., Thaker, P., Deits, R., Simeonov, D., Kollar, T., & Roy, N. (2013). Toward information theoretic human-robot dialog. *Robotics*, *32*, 409–417.

Tellex, S., Knepper, R. A., Li, A., Rus, D., & Roy, N. (2014). Asking for help using inverse semantics. In *Proceedings of robotics: Science and systems, Vol. 2*.

Tenbrink, T., Ross, R. J., Thomas, K. E., Dethlefs, N., & Andonova, E. (2010). Route instructions in map-based human-human and human-computer dialogue: A comparative analysis. *Journal of Visual Languages & Computing*, *21*(5), 292–309.

Tenorth, M., & Beetz, M. (2009). KnowRob—knowledge processing for autonomous personal robots. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 4261–4266).

Tenorth, M., & Beetz, M. (2017). Representations for robot knowledge in the KnowRob framework. *Artificial Intelligence*, *247*, 151–169.

Traum, D. R. (1994). A computational theory of grounding in natural language conversation. PhD thesis, University of Rochester, Rochester, NY.

Trott, S., & Bergen, B. (2017). A theoretical model of indirect request comprehension. In *Proceedings of the AAAI fall symposium on artificial intelligence for human-robot interaction (AI-HRI)*.

Von Ahn, L., Blum, M., Hopper, N. J., & Langford, J. (2003). Captcha: Using hard ai problems for security. In *Proceedings of the international conference on the theory and applications of cryptographic techniques* (pp. 294–311). Springer

Wielemaker, J. (1987). SWI-Prolog documentation: Prolog for the real world. http://www.swi-prolog.org. Accessed 5 Feb 2018.

Wielemaker, J., Schrijvers, T., Triska, M., & Lager, T. (2012). SWI-Prolog. *Theory and Practice of Logic Programming*, *12*(1–2), 67–96.

Williams, T. (2017a). A consultant framework for natural language processing in integrated robot architectures. *IEEE Intelligent Informatics Bulletin*, *18*(1), 10–14.

Williams, T. (2017b). Situated natural language interaction in uncertain and open worlds. PhD thesis, Tufts University.

Williams, T. (2018a). Toward ethical natural language generation for human-robot interaction. In *Late breaking report for the 13th ACM/IEEE international conference on human-robot interaction*.

Williams, T. (2018b). "Who Should I Run Over?": Long-term ethical implications of natural language generation. In *Proceedings of the 2018 HRI workshop on longitudinal human-robot teaming*.

Williams, T., & Jackson, B. (2018). A Bayesian analysis of moral norm malleability during clarification dialogues. In *Proceedings of the fortieth annual meeting of the cognitive science society*.

Williams, T., & Scheutz, M. (2015). POWER: A domain-independent algorithm for probabilistic, open-world entity resolution. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1230–1235).

Williams, T., & Scheutz, M. (2016a). A framework for resolving open-world referential expressions in distributed heterogeneous knowledge bases. In *Proceedings of the thirtieth AAAI conference on artificial intelligence (AAAI)*, pp 3598–3964.

Williams, T., & Scheutz, M. (2016b). Resolution of referential ambiguity using Dempster-Shafer theoretic pragmatics. In *Proceedings of the AAAI fall symposium on artificial intelligence for human-robot interaction (AI-HRI)*.

Williams, T., & Scheutz, M. (2017a). Referring expression generation under uncertainty: Algorithm and evaluation framework. In *Proceedings of the 10th international conference on natural language generation (INLG)*.

Williams, T., & Scheutz, M. (2017b). Resolution of referential ambiguity in human-robot dialogue using Dempster-Shafer theoretic pragmatics. In *Proceedings of robotics: science and systems*.

Williams, T., Núñez, R. C., Briggs, G., Scheutz, M., Premaratne, K., & Murthi, M. N. (2014). A Dempster-Shafer theoretic approach to understanding indirect speech acts. *Advances in Artificial Intelligence*.

Williams, T., Briggs, G., Oosterveld, B., & Scheutz, M. (2015). Going beyond command-based instructions: Extending robotic natural language interaction capabilities. In: *Proceedings of the twenty-ninth AAAI conference on artificial intelligence (AAAI)* (pp. 1387–1393).

Williams, T., Acharya, S., Schreitter, S., & Scheutz, M. (2016). Situated open world reference resolution for human-robot dialogue. In *Proceedings of the eleventh ACM/IEEE international conference on human-robot interaction (HRI)* (pp. 311–318).

Williams, T., Johnson, C., Scheutz, M., & Kuipers, B. (2017). A tale of two architectures: A dual-citizenship integration of natural language and the cognitive map. In *Proceedings of the sixteenth international conference on autonomous agents and multi-agent systems (AAMAS)*, Sao Paolo, Brazil.

Wilson, J. R., Krause, E., Scheutz, M., & Rivers, M. (2016). Analogical generalization of actions from single exemplars in a robotic architecture. In *Proceedings of the international conference on autonomous agents and multiagent systems (AAMAS)* (pp. 1015–1023).

Wyatt, J. (2005). Planning clarification questions to resolve ambiguous references to objects. In *Proceedings of the 4th IJCAI workshop on knowledge and reasoning in practical dialogue systems, Edinburgh, Scotland* (pp. 16–23).

Yazdani, F., Scheutz, M., & Beetz, M. (2017). Guidelines for improving task-based natural language understanding in human-robot rescue teams. In *Proceedings of the 2017 8th IEEE international conference on cognitive infocommunications (CogInfoCom)*, Debrecen, Hungary, accepted for publication.

Yoon, S. O., & Brown-Schmidt, S. (2013). Lexical differentiation in language production and comprehension. *Journal of Memory and Language*, *69*(3), 397–416.

Zarrieß, S., & Schlangen, D. (2016). Towards generating colour terms for referents in photographs: Prefer the expected or the unexpected? In: *Proceedings of the 9th international natural language generation conference (INLG)*.

Zettlemoyer, L. S., & Collins, M. (2012). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. It *Proceedings of the twenty-first conference on uncertainty in artificial intelligence (UAI)*.

Tom Williams is an Assistant Professor of Computer Science at the Colorado School of Mines, where he directs the Mines Interactive Robotics Research (MIRROR) Lab. Tom earned a joint Ph.D. in Computer Science and Cognitive Science from Tufts University in 2017, and in 2016 served as a visiting researcher at the Institute for Artificial Intelligence in Bremen, Germany. His research focuses on enabling and understanding natural language based human-robot interaction, especially as applied to assistive and search-and-rescue robotics. Link: https://inside.mines.edu/~twilliams

Fereshta Yazdani graduated from the Computer Science Department of the University of Bremen in 2012, earning her Dipl.-Inf. under the supervision of Prof. Dr. Rolf Drechsler in the Computer Architecture working group, with a diploma focusing on embedding reversible systems in SystemC. From 2012 to 2017, she has been working as a research assistant under the supervision of Prof. Michael Beetz, PhD at the University of Bremen's Institute for Artificial Intelligence, developing search and rescue mechanisms for mixed human-robot teams in alpine environments as part of the EU Sherpa project. Her research interests are in the area of cognitive human-robot interaction, search and rescue robotics, and knowledge representation and processing.

Prasanth Suresh is a graduate student in the Mechanical Engineering Department at Colorado School of Mines, where he works in the MIRRORLab under Dr. Tom Williams. Prasanth earned his Bachelors in Electrical and Electronics Engineering from Rajalakshmi Engineering College, India in 2015, and has worked on numerous projects relating to Embedded Systems and Robotics. His research interests include human-centered robotics, human-robot interaction, social robotics, and embedded systems.

Matthias Scheutz is a Professor in Cognitive and Computer Science in the Department of Computer Science at Tufts University. He earned a Ph.D. in Philosophy from the University of Vienna in 1995 and a Joint Ph.D. in Cognitive Science and Computer Science from Indiana University Bloomington in 1999. He has more than 250 peer-reviewed publications in artificial intelligence, natural language processing, cognitive modeling, robotics, and human-robot interaction. His current research focuses on complex cognitive robots with natural language capabilities.

Michael Beetz is a professor for Computer Science at the Faculty for Mathematics & Informatics of the University Bremen and head of the Institute for Artificial Intelligence (IAI). IAI investigates AI-based control methods for robotic agents, with a focus on human-scale everyday manipulation tasks. He is also the coordinator of the Collaborative Research Centre EASE (Everyday Activity Science and Engineering) at the University Bremen. Michael Beetz received his diploma degree in Computer Science with distinction from the University of Kaiserslautern. His MSc, MPhil, and PhD degrees were awarded by Yale University in 1993, 1994, and 1996 and his Venia Legendi from the University of Bonn in 2000. Michael Beetz Was the vice coordinator of the German Cluster of Excellence CoTeSys (Cognition for Technical Systems) and European integrating project RoboHow.