An Information-State Approach to Collaborative Reference

David DeVault[1] **Natalia Kariaeva**[2] **Anubha Kothari**[2] **Iris Oved**[3] and **Matthew Stone**[1]
[1]Computer Science [2]Linguistics [3]Philosophy and Center for Cognitive Science
Rutgers University
Piscataway NJ 08845-8020
Firstname.Lastname@Rutgers.Edu

Abstract

We describe a dialogue system that works with its interlocutor to identify objects. Our contributions include a concise, modular architecture with reversible processes of understanding and generation, an information-state model of reference, and flexible links between semantics and collaborative problem solving.

1 Introduction

People work together to make sure they understand one another. For example, when identifying an object, speakers are prepared to give many alternative descriptions, and listeners not only show whether they understand each description but often help the speaker find one they do understand (Clark and Wilkes-Gibbs, 1986). This natural collaboration is part of what makes human communication so robust to failure. We aim both to explain this ability and to reproduce it.

In this paper, we present a novel model of collaboration in referential linguistic communication, and we describe and illustrate its implementation. As we argue in Section 2, our approach is unique in combining a concise abstraction of the dynamics of joint activity with a reversible grammar-driven model of referential language. In the new information-state model of reference we present in Section 3, interlocutors work together over multiple turns to associate an entity with an agreed set of concepts that characterize it. On our approach, utterance planning and understanding involves reasoning about how domain-independent linguistic forms can be used in context to contribute to the task; see Section 4. Our system reduces to four modules: understanding, update, deliberation and generation, together with some supporting infrastructure; see Section 5. This design derives the efficiency and flexibility of referential communication from carefully-designed representation and reasoning in this simple architecture; see Section 6. With this proof-of-concept implementation, then, we provide a jumping-off point for more detailed investigation of knowledge and processes in conversation.

2 Overview and Related Work

Our demonstration system plays a referential communication game, much like the one that pairs of human subjects play in the experiments of Clark and Wilkes-Gibbs (1986). We describe each episode in this game as an activity involving the coordinated action of two participants: a *director D* who knows the referent *R* of a target variable *T* and a *matcher M* whose task is to identify *R*. Our system can play either role, *D* or *M*, using virtual objects in a graphical display as candidate targets and distractors, and using text as its input and output. Our system uses the same task knowledge and the same grammar whichever role it plays. Of course, the system also draws on private knowledge to decide how best to carry out its role; for now it describes objects using the domain-specific iteration proposed by Dale and Reiter (1995). The knowledge we have formalized is targeted to a proof-of-concept implementation, but we see no methodological obstacle in adding to the

Proceedings of the ACL Interactive Poster and Demonstration Sessions,
pages 1–4, Ann Arbor, June 2005.

system's resources.

We exemplify what our system does in (1).

(1) a. S: This one is a square.
b. U: Um-hm...
c. S: It's light brown.
d. U: You mean like tan?
e. S: Yeah.
f. S: It's solid.
g. U: Got it.

The system (S) and user (U) exchange seven utterances in the course of identifying a tan solid square.

We achieve this interaction using the information-state approach to dialogue system design (Larsson and Traum, 2000). This approach describes dialogue as a coordinated effort to maintain an agreed record of the state of the conversation. Our model contrasts with traditional plan-based models, as exemplified by Heeman and Hirst's model of goals and beliefs in collaborative reference (1995). Our approach abstracts away from such details of individuals' mental states and cognitive processes, for principled reasons (Stone, 2004a). We are able to capture these details *implicitly* in the dynamics of conversation, whereas plan-based models must represent them *explicitly*. Our representations are simpler than Heeman and Hirst's but support more flexible dialogue. For example, their approach to (1) would have interlocutors coordinating on goals and beliefs about a syntactic representation for *the tan solid square*; for us, this description and the interlocutors' commitment to it are abstract results of the underlying collaborative activity.

Another important antecedent to our work is Purver's (2004) characterization of clarification of names for objects and properties. We extend this work to develop a treatment of referential descriptive clarification. When we describe things, our descriptions grow incrementally and can specify as much detail as needed. Clarification becomes correspondingly cumulative and open-ended. Our revised information state includes a model of cumulative and open-ended collaborative activity, similar to that advocated by Rich et al. (2001). We also benefit from a reversible goal-directed perspective on descriptive language (Stone et al., 2003).

3 Information State

Our information state (IS) models the ongoing collaboration using a stack of tasks. For a task of collaborative reference, the IS tracks how interlocutors together set up and solve a constraint-satisfaction problem to identify a target object. In any state, *D* and *M* have agreed on a target variable *T* and a set of constraints that the value of *T* must satisfy. When *M* recognizes that these constraints identify *R*, the task ends successfully. Until then, *D* can take actions that contribute new constraints on *R*. Importantly, what *D* says adds to what is already known about *R*, so that the identification of *R* can be accomplished across multiple sentences with heterogeneous syntactic structure.

Our IS also allows subtasks of questioning or clarification that interlocutors can use to maintain alignment. The same constraint-satisfaction model is used not only for referring to displayed objects but also for referring to abstract entities, such as actions or properties. Our IS tracks the salience of entity and property referents and, like Purver's, maintains the previous utterance for reference in clarification questions. Note, however, that we do not factor updates to the IS through an abstract taxonomy of speech acts. Instead, utterances directly make domain moves, such as adding a constraint, so our architecture allows utterances to trigger an open-ended range of domain-specific updates.

4 Linguistic Representations

The way utterances signal task contributions is through a collection of presupposed constraints. To understand an utterance, we solve the utterance's grammatically-specified semantic constraints. An interpretation is only feasible if it represents a contextually-appropriate contribution to the ongoing task. Symmetrically, to generate an utterance, we use the grammar to formulate a set of constraints; these constraints must identify the contribution the system intends to make. We view interpreted linguistic structures as representing communicative intentions; see (Stone et al., 2003) or (Stone, 2004b).

As in (DeVault et al., 2004), a *knowledge interface* mediates between domain-general meanings and the domain-specific ontology supported in a particular application. This allows us to build inter-

pretations using domain-specific representations for referents, for task moves, and for the domain properties that characterize referents.

5 Architecture

Our system is implemented in Java. A set of interface types describes the flow of information and control through the architecture. The representation and reasoning outlined in Sections 3 and 4 is accomplished by implementations of these interfaces that realize our approach. Modules in the architecture exchange messages about events and their interpretations. (1) Deliberation responds to changes in the IS by proposing task moves. (2) Generation constructs collaborative intentions to accomplish the planned task moves. (3) Understanding infers collaborative intentions behind user actions. Generation and understanding share code to construct intentions for utterances, and both carry out a form of inference to the best explanation. (4) Update advances the IS symmetrically in response to intentions signaled by the system or recognized from the user; the symmetric architecture frees the designer from programming complementary updates in a symmetrical way. Additional supporting infrastructure handles the recognition of input actions, the realization of output actions, and interfacing between domain knowledge and linguistic resources.

Our system is designed not just for users to interact with, but also for demonstrating and debugging the system's underlying models. Processing can be paused at any point to allow inspection of the system's representations using a range of visualization tools. You can interactively explore the IS, including the present state of the world, the agreed direction of the ongoing task, and the representation of linguistic distinctions in salience and information status. You can test the grammar and other interpretive resources. And you can visualize the search space for understanding and generation.

6 Example

Let us return to dialogue (1). Here the system represents its moves as successively constraining the shape, color and pattern of the target object. In generating (1c), the system iteratively elaborates its description from *brown* to *light brown* in an attempt to identify the object's color unambiguously. The user's clarification request at (1d) marks this description of color as problematic and so triggers a nested instance of the collaborative reference task. At (1e) the system adds the user's proposed constraint and (we assume) solves this nested subtask. The system returns to the main task at (1f) having grounded the color constraint and continues by identifying the pattern of the target object.

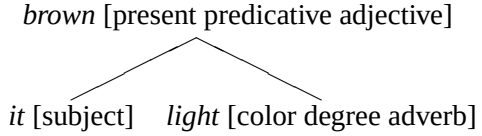
Let us explore utterance (1c) in more detail. The IS records the status of the identification process. The system is the director; the user is the matcher. The target is represented provisionally by a discourse referent t_1, and what has been agreed so far is that the current target is a square of the relevant sort for this task, represented in the agent as *square-figure-object*(t_1). In addition, the system has privately recorded that square o_1 is the referent it must identify. For this IS, it is expected that the director will propose an additional constraint identifying t_1. The discourse state represents t_1 as being *in-focus*, or available for pronominal reference.

Deliberation now gives the generator a specific move for the system to achieve:

(2) *add-constraint*$(t_1, $*color-sandybrown*$(t_1))$

The content of the move in (2) is that the system should update the collaborative reference task to include the constraint that the target is drawn in a particular, domain-specific color (RGB value F4-A4-60, or XHTML standard "sandy brown"). The system finds an utterance that achieves this by exploring head-first derivations in its grammar; it arrives at the derivation of *it's light brown* in (3).

(3)



A set of presuppositions connect this linguistic structure to a task domain; they are given in (4a). The relevant instances in this task are shown in (4b).

(4) a. *predication*$(M) \wedge$ *brown*$(C) \wedge$ *light*(C)

b. *predication*$($*add-constraint*$) \wedge$
brown$($*color-sandybrown*$) \wedge$
light$($*color-sandybrown*$)$

The utterance also uses *it* to describe a referent X so presupposes that *in-focus*(X) holds. The move effected by the utterance is schematized as $M(X,C(X))$. Given the range of possible task moves in the current context, the constraints specified by the grammar for (3) are modeled as determining the instantiation in (2). The system realizes the utterance and assumes, provisionally, that the utterance achieves its intended effect and records the new constraint on t_1.

Because the generation process incorporates entirely declarative reasoning, it is normally reversible. Normally, the interlocutor would be able to identify the speaker's intended derivation, associate it with the same semantic constraints, resolve those constraints to the intended instances, and thereby discover the intended task move. In our example, this is not what happens. Recognition of the user's clarification request is triggered as in (Purver, 2004). The system fails to interpret utterance (1d) as an appropriate move in the main reference task. As an alternative, the system "downdates" the context to record the fact that the system's intended move may be the subject of explicit grounding. This involves pushing a new collaborative reference task on the stack of ongoing activities. The system remains the director, the new target is the variable C in interpretation and the referent to be identified is the property *color-sandybrown*. Interpretation of (1d) now succeeds.

7 Discussion

Our work bridges research on collaborative dialogue in AI (Rich et al., 2001) and research on pragmatics in computational linguistics (Stone et al., 2003). The two traditions have a lot to gain from reconciling their assumptions, if as Clark (1996) suggests, people's language use is coextensive with their joint activity. There are implications both ways.

For pragmatics, our model suggests that language use requires collaboration in part because reaching agreement about content involves substantive social knowledge and coordination. Indeed, we suspect that collaborative reference is only one of many relevant social processes. For collaborative dialogue systems, adopting rich declarative linguistic representations enables us to directly interface the core modules of a collaborative system with one another. In language understanding, for example, we can collapse together notional subprocesses like semantic reconstruction, reference resolution, and intention recognition and solve them in a uniform way.

Our declarative, reversible approach supports an analysis of how the system's specifications drive its input-output behavior. The architecture of this system thus provides the groundwork for further investigations into the interaction of social, linguistic, cognitive and even perceptual and developmental processes in meaningful communication.

Acknowledgements

Supported in part by NSF HLC 0308121. Thanks to Paul Tepper.

References

H. H. Clark and D. Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22:1–39.

H. H. Clark. 1996. *Using Language*. Cambridge.

R. Dale and E. Reiter. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 18:233–263.

D. DeVault, C. Rich, and C. L. Sidner. 2004. Natural language generation and discourse context: Computing distractor sets from the focus stack. In *FLAIRS*.

P. Heeman and G. Hirst. 1995. Collaborating on referring expressions. *Comp. Ling.*, 21(3):351–382.

S. Larsson and D. Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Eng.*, 6:323–340.

M. Purver. 2004. *The Theory and Use of Clarification Requests in Dialogue*. Ph.D. thesis, Univ. of London.

C. Rich, C. L. Sidner, and N. Lesh. 2001. COLLAGEN: applying collaborative discourse theory to human-computer interaction. *AI Magazine*, 22:15–25.

M. Stone, C. Doran, B. Webber, T. Bleam, and M. Palmer. 2003. Microplanning with communicative intentions. *Comp. Intelligence*, 19(4):311–381.

M. Stone. 2004a. Communicative intentions and conversational processes. In J. Trueswell and M. K. Tanenhaus, editors, *Approaches to Studying World-Situated Language Use*, pages 39–70. MIT.

M. Stone. 2004b. Intention, interpretation and the computational structure of language. *Cognitive Science*, 28(5):781–809.