Advances in Cognitive Systems 3 (2014) 201–219 Submitted 9/2013; published 7/2014

Integrating Meta-Level and Domain-Level Knowledge for Task-Oriented Dialogue

Alfredo Gabaldon ALFREDO.GABALDON@SV.CMU.EDU
Pat Langley PATRICK.W.LANGLEY@GMAIL.COM
Silicon Valley Campus, Carnegie Mellon University, Moffett Field, CA 94035 USA

Ben Meadows BMEA011@AUCKLANDUNI.AC.NZ
Department of Computer Science, University of Auckland, Private Bag 92019, Auckland 1142 NZ

Abstract

There is general agreement that knowledge plays a key role in intelligent behavior, but most work on this topic has emphasized domain-specific expertise. We argue, in contrast, that cognitive systems also benefit from meta-level knowledge that has a domain-independent character. In this paper, we propose a representational framework that distinguishes between these two forms of content, along with an integrated architecture that supports their use for abductive interpretation and hierarchical skill execution. We demonstrate this framework's viability on high-level aspects of extended dialogue that require reasoning about, and altering, participants' beliefs and goals. Furthermore, we demonstrate its generality by showing that the meta-level knowledge operates with different domain-level content. We conclude by reviewing related work on these topics and discussing promising directions for future research.

1. Introduction

Cognitive systems, whether human or artificial, depend on knowledge for their operation. Studies of expertise in both psychology and AI have repeatedly shown the power of domain-specific content. Experts in game playing, medical diagnosis, physics problem solving, and engineering design all draw on knowledge about their area to find better solutions, with less effort, than novices. One common inference is that researchers interested in developing cognitive systems should focus their energies on constructing domain-specific encyclopedias, as opposed to finding general principles of intelligence. Another is that general strategies about how to use this domain-specific knowledge are best encoded in procedural rather than declarative terms.

We argue that both conclusions are flawed. We maintain that, although domain-specific content plays an important role in intelligent behavior, it also relies on meta-level knowledge that generalizes across many different domains, and that much of this knowledge encodes strategic information. In this paper, we apply these ideas to high-level aspects of dialogue, which we hold depend on both domain-level and meta-level processing. Conversations are always about some topic, which requires the use of domain expertise, but they also involve more abstract structures that reflect the general character of dialogues. In this paper, we report an integrated architecture that incorporates this distinction by cleanly separating domain-level from meta-level content, and that uses them both during conceptual inference to interpret the goals and beliefs of participants, as well as during the execution of skills to achieve its goals. We then demonstrate both ideas in the context of a dialogue system that operates across multiple domain predicates.

In the next section, we present a motivating example that illustrates the basic ideas. After this, we discuss the architecture's representation of domain-level and meta-level content, followed by mechanisms that combine them to draw abductive inferences about the participants' mental states and to carry out hierarchical, goal-directed activities. Although we explore these ideas in the context of dialogue, our system does not attempt to model speech or sentence processing. Rather, it assumes the logical meanings of utterances as inputs and outputs, supporting both dialogue interpretation and generation at this abstract level. Moreover, we believe the same approach will prove useful in other settings that involve social cognition, although we do not offer empirical evidence for that here. We hold that one can specify meta-level knowledge that is useful across different domain-specific content and that dialogue is one area that highlights this idea. To this end, we demonstrate the architecture's operation on conversations about different topics that involve distinct predicates and domain content, but that use the same meta-level rules. We also examine related work that touches on many of the ideas we have incorporated in our architecture.

We will see that each of the ideas we have adopted – the distinction between domain-level and meta-level structures, the importance of hierarchical knowledge organization, mental models of other agents' beliefs and goals, and reliance on abduction to infer them – have all been explored in the literature, and we have borrowed freely from the earlier publications that introduced them. We will not claim that we are the first to propose any of these techniques, but, to our knowledge, *our architecture provides the first integration of them for interpreting and generating dialogue about complex goal-directed activities*. Langley (2012) has argued that such integration efforts are a hallmark of cognitive systems research, and we view our main contribution in that light.

2. A Motivating Example

We can clarify the issues we intend to address with a motivating example of a task-oriented dialogue in which the interacting agents adopt and pursue a shared goal. We assume that the participants cooperate because they cannot achieve the goal individually. Consider a situation in which one agent – a medic in a battlefield – is assisting an injured person and seeks the advice of another agent – the advisor. The latter has the expertise needed to solve the problem, but cannot affect the environment directly, whereas the medic has limited training and expertise but can act if provided with appropriate instruction.

The idealized dialogue shown in Table 1, despite its simplicity, raises many of the research issues that we want to address. Here are some important but noncontroversial observations about characteristics of the dialogue:

- Behavior is goal-directed and involves joint activity over time; this activity includes not only domain actions carried out by the caller, but communicative actions, including ones for information gathering, by both parties;

Table 1. Sample dialogue involving joint activity between a medic and an advisor.

Medic:	We have a man injured!
Advisor:	Ok. What type of injury?
Medic:	He's bleeding.
Advisor:	How bad is the bleeding?
Medic:	Pretty bad. I think it is an artery.
Advisor:	Ok. Where is the injury?
Medic:	It's on the left leg.
Advisor:	Apply pressure on the leg's pressure point.
Medic:	Roger that.
Advisor:	Has the bleeding stopped?
Medic:	No. He's still bleeding.
Advisor:	Ok. Apply a tourniquet.
Medic:	Where do I put the tourniquet?
Advisor:	Just below the joint above the wound.
Medic:	Ok. The bleeding has stopped.

- Participants develop a shared model not only of the environmental situation, but of each others' beliefs and goals; this joint model, or *common ground* (Clark, 1996), gradually expands as the participants exchange more information;
- Many of the agents' beliefs and goals are never stated but are nevertheless inferred by the other participant; some inferences involve domain content, but others concern communication-related goals and beliefs;
- The overall process, from the perspective of either agent, alternates between drawing inferences, based on the other's utterance, to expand his or her own understanding of the situation and carrying out goal-directed activities in response to the resulting model.

These observations act as requirements about the abilities of any computational system that we might develop to mimic this behavior. These include the ability to incrementally construct an interpretation of both the physical situation and of others' goals and beliefs, to carry out goal-directed activities appropriate to this inferred model, and to utilize domain knowledge and more abstract structures for both inference and execution. Moreover, it seems likely that the same abstract content would apply if the domain situation were different, suggesting the need for generalized meta-level knowledge and for mechanisms to process it.

In the next section, we describe the representations and processes of an integrated architecture that responds to these requirements, after which we provide preliminary evidence for the generality of our approach. Although we illustrate the architecture's assumptions in the context of task-oriented dialogue, the framework itself is considerably more general and should be relevant to any setting that involves social cognition, that is, to any reasoning and activity that incorporates models of other agents' mental states.

3. An Architecture for Meta-Level Cognition

We desire an architectural framework that supports goal-directed but informed behavior over time. This should handle the higher levels of task-oriented dialogue but also be general enough to support other varieties of social cognition. We take as our role model ICARUS (Langley, Choi, & Rogers, 2009), a cognitive architecture designed for physical agents. In this section, we describe the new framework's memories, representations, and mechanisms, noting the ways in which it differs from ICARUS to incorporate models of other agents' mental states and to handle meta-level processing.

3.1 Working Memory

The new architecture stores beliefs and goals in a dynamic working memory that contains ground literals. In contrast to ICARUS, which has distinct memories for beliefs and goals, the new framework includes both in a single memory, distinguishing them with the meta-level predicates *belief* and *goal*. These take the form

$$\textit{belief}(A, C) \quad \text{and} \quad \textit{goal}(A, C)\ ,$$

respectively, where A denotes an agent and C encodes domain-level content, an embedded belief or goal expression, or an expression denoting the occurrence of a communicative act.

Domain-level literals encode information related to the domain. These always occur as arguments of some meta-level predicate like *belief* or *goal*; they never appear at the top level of working memory. Although these may be stated in the notation of predicate logic, here we will assume they take the form of *triples*, which lets the architecture access domain-level predicates as easily as their arguments. For instance, rather than storing the element $\textit{belief}(\textit{Advisor}, \textit{injured}(\textit{p1}))$, it would store the equivalent elements

$$\textit{belief}(\textit{Advisor}, [\textit{inj1, type, injury}]),$$
$$\textit{belief}(\textit{Advisor}, [\textit{inj1, agent, p1}]).$$

Note that this encoding also reifies the event or relation, which serves as the anchor point for a set of relational triples that describe that event or relation.

The architecture also allows meta-level expressions of the form $not(C)$, which denotes the negation of some belief or goal C, as well as expressions that have the forms *belief_if(A, C)*, where A refers to an agent and C to a domain-level expression that A knows or does not know, respectively. Similarly, for a domain-level, binary relation R we use $\textit{belief\_wh}(A, [X, R])$ to state that agent A *believes what* value X takes for the relation$R$. For instance,

$$\textit{belief}(A, \textit{belief\_wh}(B, [\textit{inj1, injury\_type}]))$$

represents A's belief that B knows the type of injury $inj1$ has, such as bleeding. Taken together, these meta-level predicates let the architecture encode not only its beliefs and goals about the environment, but also its beliefs and goals about other agents' mental states, including things they do and do not know.

Because our current application involves dialogue, we will also assume meta-level predicates for *speech acts* (Austin, 1962; Searle, 1969), although they are not strictly part of the architecture.

These denote conversational actions that a participant performs to produce certain effects on others. Different researchers have proposed alternative taxonomies of speech acts, but here we adopt a minimal set of six types that appear sufficient for our purposes:

- *inform*(S, L, C); in which speaker S asks listener L to believe content C;
- *acknowledge*(S, L, C); in which S tells L he has received and now believes content C;
- *question*(S, L, C); in which S asks L a question C;
- *propose*(S, L, C); in which S asks L to adopt goal C;
- *accept*(S, L, C); in which S tells L he has adopted goal C;
- *reject*(S, L, C); in which S tells L he has rejected goal C.

In working memory, speech act expressions always appear as the content of beliefs or goals. For instance, the literal *belief*$(A,$ *inform*$(B, A,$ *Content*$))$ encodes A's belief that a speech act has occurred in which B informed A about *Content*.

3.2 Conceptual and Skill Knowledge

Like ICARUS, the new architecture incorporates generic knowledge that lets it interpret and alter the contents of working memory. This includes concepts that support inference and skills that enable goal-directed activity. We can partition these structures into domain-level and meta-level knowledge. By *meta-level knowledge*, we mean knowledge that does not refer to domain content directly, but instead incorporates meta-level predicates that take such content as arguments, with the latter's values instantiated at processing time.

Domain-level concepts are specified by rules in which the head specifies the conceptual predicate and its arguments, and in which the body comprises a relational pattern associated with that concept. The head may include multiple literals, which is required by the triples representation. Domain-level concepts do not mention an agent, since they are always relative to a single agent's beliefs or goals. As in ICARUS, conceptual knowledge is organized in a hierarchy, with more complex predicates defined in terms of more basic ones.

Domain-level skills describe the effects of an agent's activities under certain conditions. Like concepts, they are specified by rules with a head that includes the skill's predicate and arguments, and a body that includes conditions which must be satisfied for it to execute. The bodies of primitive skills refer to executable actions, whereas nonprimitive skills include a sequence of lower-level skills. For example, the nonprimitive skill

$$\begin{aligned}
&\textit{skill}([P, \textit{status}, \textit{stable}]) \leftarrow \\
&\qquad \textit{belief}(A, [\textit{Inj}, \textit{agent}, P]), \\
&\qquad \textit{belief}(A, [\textit{Inj}, \textit{type}, \textit{injury}]), \\
&\qquad \textit{skill}(\textit{belief\_wh}(A, [\textit{Inj}, \textit{injury\_type}])), \\
&\qquad \textit{skill}([[\textit{Inj,injury\_type,InjType}], [P, \textit{status}, \textit{stable}]])).
\end{aligned}$$

states that, to stabilize an injured person, one should first determine the type of the injury (e.g., by asking a question), and then carry out a subskill appropriate for that type of injury. Different rules with the same head denote alternative ways to decompose a skill. Thus, as in the ICARUS architecture, the set of skill clauses is equivalent to a hierarchical task network.

Goal-generating rules constitute a third type of domain-level knowledge. These differ from concepts and skills in that they do not define new predicates, but rather describe the conditions under which one should establish a top-level goal. For example, an agent might have a rule that, when it believes a person is in a life-threatening condition, creates the goal of stabilizing that person. This rule is stated

$$\textit{goal}(Agent, [P, status, stable]) \leftarrow \quad \textit{belief}(\textit{Agent}, [P, \textit{condition}, \textit{life\_threatening}]).$$

Our approach is very similar to the extension of ICARUS that Choi (2010) reported. In both cases, rules generate goals about desired states that in turn lead the architecture to select skills that refer to them in their heads.

However, the new architecture's support for meta-level knowledge is another major departure from ICARUS. In the context of dialogue, the key meta-level concepts are encoded by rules for speech acts. For example, the rule for the *propose* speech act, from the speaker's perspective, can be stated

$$\begin{aligned} \textit{propose}(S, L, C) \leftarrow \quad & \textit{goal}(S, C), \\ & \textit{goal}(S,\ \textit{goal}(L, C)), \\ & \textit{belief}(S, \textit{goal}(L, C)). \end{aligned}$$

This means that the act of speaker S proposing content C to listener L is associated with S having C as a goal, S having a goal for L to adopt C as a goal, and S believing that L has adopted C as a goal. The reader may find this rule somewhat strange. Although it is meant to capture the effects of the speech act under certain conditions, the rule does not differentiate between the conditions $\textit{goal}(S, C)$ and $\textit{goal}(S,\ \textit{goal}(L, C))$ and the effect $\textit{belief}(S, \textit{goal}(L, C))$. Simplified rules of this sort are sufficient for our system, which builds explanations only by adding newly inferred content. In a more general setting, however, it will not suffice. In Section 3.4, we consider how one might adapt such rules to incorporate temporal information.

Different speech acts involve different patterns of goals and beliefs, but none refer to any domain predicates, since the content of the speech act does not alter the abstract relations. Our dialogue system incorporates 14 rules of this sort, one for each type of speech act for the speaker's perspective and one each for the listener, except for the question speech act. A question requires four rules: two for questions using a triple $[X, R, Y]$ as content and two for questions with content $[X, R]$. The former are if questions that are answered according to whether the listener believes the triple $[X, R, Y]$ or not. The latter are wh questions that are answered with a complete triple $[X, R, Y]$, providing the value Y as the answer to the wh question if the listener believes $[X, R, Y]$.

Another form of meta-level knowledge is a *dialogue grammar* that specifies relations among speech acts. This includes hierarchical and recursive rules that state, for instance, a dialogue may consist of S asking L a question Q, followed by L answering S with A, followed by another dialogue. As with speech act rules, those for the dialogue grammar make no reference to any domain predicates. Of course, to ensure a coherent conversation, the system must examine the domain content to check that it makes sense. To this end, the system includes another four meta-level predicates that indicate 'conceptual agreement' between the arguments of different pairs of

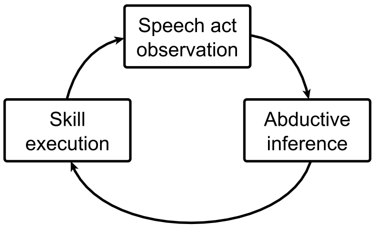

Figure 1. The architecture's cognitive cycle includes an observation step that inputs speech acts, an abductive inference step that extends the current explanation, and an execution step that selects and carries out a skill relevant to the current working memory.

speech acts, along with 13 meta-level rules that determine whether they are satisfied. For example, two rules for wh questions ensure that answers are consistent with the agent's beliefs:

$$\textit{que\_inf\_p}(\textit{question}(S, L, [X, R]), \textit{inform}(L, S, [X, R, Y])) \leftarrow \ \textit{belief}(L, [X, R, Y]).$$

$$\begin{aligned}\textit{que\_inf\_p}(\textit{question}(S, L, [X, R]), \textit{inform}(L, S, neg(\textit{belief\_wh}(L, [X, R])))) \leftarrow \\ \textit{belief}(L, neg(\textit{belief\_wh}(L, [X, R]))).\end{aligned}$$

The first rule says that a wh question must be answered with a complete triple whose first two elements match those in the question, and the answering agent must believe this triple. The second rule states the condition for answering "I do not know" to a wh question, where the expression $neg(\textit{belief\_wh}(L, [X, R]))$ means that the agent does not believe $[X, R, Y]$ for any value Y.

This completes our description of the architecture's central representational commitments and their use in the context of dialogue. Next we turn to the computational mechanisms that operate over these cognitive structures, including their mapping onto processes for dialogue interpretation and generation.

3.3 Conceptual Inference and Skill Execution

Following ICARUS, the new architecture operates in distinct cognitive cycles, with the first stage of each cycle involving conceptual inference over the agent's observations. Because ICARUS deals primarily with physical environments, it could rely on deduction to draw conclusions about its immediate situation. However, social cognition, including that arising in dialogue, requires making plausible assumptions about unobserved mental states. For this reason, the architecture utilizes an abductive inference mechanism that attempts to explain the observations to date in terms of available knowledge. This process operates incrementally, in that it extends the explanation produced on the previous cycle, adding new elements to working memory monotonically. The abduction process prefers the explanation that requires the fewest default assumptions, which it finds through an iterative process.

The module first attempts to prove some top-level relation (e.g., that the observed speech acts form a dialogue) with no assumptions, then considers accounts that require one assumption, then two assumptions, and so on, continuing until it finds a complete proof or until it exceeds a limit.

Once the architecture finds an explanation, it adds the default assumptions to working memory, treating these elements as givens on the next round, so that typically it must introduce only a few new assumptions per cycle. The current implementation does not support belief revision, which is clearly needed in cases of misunderstanding, but it has worked well in practice in the scenarios we have examined.

When applied to dialogue, the abduction mechanism attempts to incorporate 'observed' speech acts that are added to working memory after each participant has taken his turn speaking. The system introduces beliefs and goals of the two agents as default assumptions, with rules for speech acts providing the lower layer of the proof tree and dialogue grammar rules generating the higher levels. Omitted speech acts, such as implicit acknowledgments, cause no difficulty, since they are also introduced as default assumptions. These become working memory elements and thus can serve as terminal nodes in the expanded explanation on successive cycles.

Once the architecture has completed the conceptual inference stage, it matches the conditions of all goal-generation rules against the updated contents of working memory. Each rule that matches adds a new top-level goal to working memory, instantiating the arguments of its predicate with bindings obtained during the match process. These goals describe aspects of a desired state that appear in the heads of relevant skills. In addition, abductive inference may introduce new top-level goals, as when an agent proposes an objective and there is no evidence the listener has rejected it.

The new architecture also shares with ICARUS a skill execution stage that immediately follows conceptual inference and goal generation. The execution process selects some top-level goal and then finds a skill clause with this goal in its head and with conditions that match working memory. The module repeats this step recursively to select a path downward through the skill hierarchy that is relevant to achieving the top-level goal. Upon reaching a primitive skill, the architecture replaces variables with their bindings and carries out its associated action.

On the next cycle, if the system selects the same top-level goal, it repeats this process. However, if the conditions of some skills along the previous path are no longer satisfied, the execution module may follow a slightly different route. This can lead the agent to carry out subskills in sequence, from left to right, as each one completes its activity. As in ICARUS, this gives the architecture a reactive flavor, although the influence of the top-level goal also provides it with continuity. The result is behavior similar to that of a hierarchical task network, with the system traversing different branches of an AND tree across successive cycles.

In the context of dialogue, the execution mechanism produces different behavior depending on whether the selected top-level goal was produced by abductive inference or by domain-level goal generation. In the former case, the agent may adopt a new goal like

$$goal(medic, \textit{belief}(advisor, [\textit{inj1,injury\_type,bleeding}]))\ ,$$

which could lead the architecture to access a meta-level skill that invokes the *inform* speech act. However, obtaining this answer might in turn require domain-level activities for information gathering. This operating style is common when an incoming speech act requires some response. In contrast, a domain-level goal can, in the process of executing domain-specific skills, lead to situations in which the agent requires assistance. In response, execution may invoke a meta-level skill for asking a question or otherwise obtaining aid.

We have implemented the new architecture in Prolog, which supports the embedded structures that are central to our extended formalism, as well as the pattern matching needed during abductive inference, goal generation, and skill execution. However, the control mechanisms themselves diverge substantially from those built into Prolog. The abductive inference module constructs explanations that depend on default assumptions, whereas skill execution halts after traversing a single path per cycle, rather than expanding an entire AND tree. The resulting behavior is much closer to that found in ICARUS, as seems appropriate for an architecture that carries out activities over time.

3.4 Representing and Reasoning About Time

As noted earlier, our rules for describing speech acts lack details about their elements' temporal ordering. A fuller account of dialogue processing would encode the fact that some elements hold before an utterance and others become true only afterwards. We should consider briefly how one might elaborate our representation and reasoning methods to incorporate such timing information.

We can extend the formalism by adding two arguments to literals that specify their start times and end times, with the interval indicating when the relation holds. For beliefs and goals, the start times indicate when an agent begins to believe content C or adopts C as a goal; similarly, the end times denote the point at which the agent ceases to believe C or abandons it as a goal. In this more expressive notation, the rule for *propose* would be

$$\begin{aligned} \textit{propose}(S, L, C, S0, E0) \leftarrow \quad & \textit{goal}(S, C, S1, E1), \\ & \textit{goal}(S,\ \textit{goal}(L, C, S4, E4), S2, E2), \\ & \textit{belief}(S, \textit{goal}(L, C, S5, E5), S3, E3), \\ & S1 < S0,\ S2 < S0, \\ & E0 < S3, E0 < S4,\ E0 < S5. \end{aligned}$$

Constraints among these temporal arguments capture the causal ordering between the speech acts' conditions and effects. For instance, the start time $S3$ of effect $\textit{belief}(S, \textit{goal}(L, C, S5, E5), S3, E3)$ must come after the end time $E0$ of the *propose* utterance. These constraints specify that some relations hold only before the activity, that different ones become true only afterward, and that still others hold throughout. Such temporal orderings may appear even when the particular time stamps are unknown, in which case Skolem constants are attached to the time points instead of giving them fixed values. Because times can be unknown but nevertheless constrained, we must also include constraint instances that persist in working memory to encode relationships an agent maintains among the time stamps for beliefs, goals, and other constraints.

Reasoning about such temporal constraints could occur in three contexts. First, one must maintain consistency during the interpretation process, performing simple checks for individual constraint validity when time stamps are instantiated and making more sophisticated checks to determine if sets of constraints are satisfiable when times are Skolemized. Second, when finding paths through the skill hierarchy during execution, one can reject some alternatives if existing constraints would keep their conditions from being satisfied. Third, similar reasoning can limit the application of goal-generating rules. Although we will see that the current architecture fares well without explicit temporal information, such extensions would make its representation and its mechanisms even more powerful.

4. Evidence for Generality

Our main claim is that meta-level knowledge is useful across distinct domain-specific knowledge bases and that our new architecture's mechanisms operate effectively over both forms of content. In this section, we show evidence for this generality by demonstrating its behavior on dialogues that involve different domain predicates but use the same meta-level knowledge. In the process, we provide more details about the architecture's operation on task-oriented dialogues. We have tested our system on four different domains. Here we describe in some detail the tests on two domains, one involving medical advice like the one discussed earlier and the other involving a scenario in which the system helps an elderly person track her medications.[1]

Our experimental protocol provides the system with access to a text file that contains a sequence of speech acts for the medic who is getting advice from an expert. These are instances of the speech act types described in Section 3, interleaved with two special speech acts, $over$ and out, that we use to indicate when the speaker has completed his turn and ended the conversation, respectively. The system reads the contents of this file, adding to working memory all speech acts before the next $over$. It runs for a number of cycles, continuing until it executes its own speech acts, and then accesses the file again. Of course, this protocol only makes sense if our agent responds in reasonable ways to the listed speech acts, but it provides a reliable way of testing in such cases.

In the medic scenario, the system acts as an advisor who is helping another agent, the medic, deal with a wounded person. We describe the domain using a dozen conceptual predicates like *injury* and *bleeding*. The domain knowledge includes one goal-generating rule (to stabilize an injured person) and four conceptual rules. The latter four rules infer that an injured person is stable if his wound is no longer bleeding, whether an injury is located on a limb or in the torso, and the appropriate location of a tourniquet. The domain knowledge also includes 15 skills, including

$$\begin{aligned}
&\mathit{skill}(\ [[\mathit{Inj,\ injury\_type,\ bleeding}], [\mathit{Inj,\ state,\ treated}]]\ ) \leftarrow\\
&\qquad \mathit{self}(A),\\
&\qquad \mathit{belief}(A, not(\mathit{occurred}(\mathit{skill}([[\mathit{Inj,\ injury\_type,\ bleeding}], [\mathit{Inj,\ state,\ treated}]])))),\\
&\qquad \mathit{skill}(\mathit{belief\_wh}(A, [\mathit{Inj,\ extent}])),\\
&\qquad \mathit{skill}(\ [[\mathit{Inj,\ extent,\ Ext}], [\mathit{Inj,\ injury\_type,\ bleeding}], [\mathit{Inj,\ state,\ treated}]]\ ).
\end{aligned}$$

This skill encodes an interaction in which a bleeding injury is handled by executing a subskill to determine its severity (asking the listener about its extent, as in the Section 2 dialogue) and then executing another subskill to deal with the injury accordingly. This example is interesting because a domain skill uses a meta-level skill to ask a question to satisfy a *belief_wh* goal, providing a point of interaction between the two levels.

Let us examine the system's behavior on the previous scenario. Upon observing a speech act that informs it the medic has an injured person, the system applies abductive inference and updates its working memory by adding a belief that the inform speech act occurred, that the medic has encountered an injured person, that he has a goal for the advisor to believe this, and that the advisor now believes he has an injured person. The system also generates and adds to working memory the

[1] By 'system', we mean the entire computational artifact, including the architecture and the knowledge we provide it at both the meta and domain levels.

top-level goal of stabilizing the injured person,

$$goal(\textit{advisor}, [\textit{p1, status, stable}])\ .$$

Next the skill execution process by producing an *acknowledge* speech act, such as

$$acknowledge(\textit{advisor, medic}, [\textit{inj1, agent, p1}])\ ,$$

after which it invokes a high-level skill for stabilizing the injured person. This leads it to produce a *question* speech act that asks the medic about the type of injury,

$$question(\textit{advisor, medic}, [\textit{inj1, injury\_type}])\ .$$

Note that this is a wh-question with content of the form $[X, R]$. The advisor expects an answer with a complete triple as content, such as [*inj1, injury_type, bleeding*].

The dialogue then continues with an exchange of similar questions and answers about the injury. From the medic's answers, the system infers that the first attempt at stopping the bleeding did not work and it instructs the medic to use a tourniquet instead. After further questions and instructions, the medic informs the advisor that the bleeding is under control and the interaction ends. Throughout the conversation, the system uses its dialogue grammar to update its beliefs and goals, and it draws on meta-level rules about speech acts and conceptual agreement to generate responses. The contents of the working memory influences the direction the dialogue takes by affecting which skills are applicable. For instance, if the medic answers the advisor's early question about the bleeding having stopped in the affirmative, then the system does not pursue any further skills related to this objective, changing the conversational path. Table 2 shows the new beliefs and goals that the system adds to working memory after each of the utterances in a segment of this dialogue.

The system's knowledge lets it respond adaptively to variations in the dialogue. For instance, one way in which the dialogue may follow a different direction occurs when the medic provides different information, say that the injury is a bone fracture instead of a bleeding wound, which leads the system to execute different skills and provide different advice to the medic. However, the dialogue can also vary at the higher level of speech acts. For instance, a *propose* speech act may be followed by a *reject* speech act instead of an *accept*. In the medic scenario, this can occur, for example, if the medic rejects the proposal to apply a tourniquet because it does not have one, which would take the dialogue in a different direction.

Our second domain involves an elderly person who the system attempts to assist in keeping track of medication. This setting is somewhat less complex, involving only five domain relations. Here we provide one goal-generating rule that is related to getting the person to take his medication, along with 12 domain-level skills. In this scenario, the system initially believes that it is time for the elder to take his medicine, so the goal-generation process immediately adds a top-level goal to this end, which in turn leads to execution of a relevant skill. The dialogue starts with the system taking the initiative to inform the person it is time to take his medicine, which he acknowledges, after which it informs the person that he needs a glass of water and proposes getting one.

Next the system informs the elder that he needs his pills and asks whether he knows their location. The person responds that he left the pills in the living room, so it proposes that he get them. Finally, the system reminds the person to take his pills, he responds that he has done so, and the

Table 2. Partial trace of the medic dialogue showing the content added to the working memory after each utterance. The symbol A stands for the advisor and M for the medic.

(1) **Medic: We have a man injured!**
belief(A,inform(M,A,[p1,type,person]))
belief(A,[p1,type,person])
belief(A,goal(M,belief(A,[p1,type,person])))
belief(A,belief(M,[p1,type,person]))
belief(A,inform(M,A,[inj1,type,injury]))
belief(A,[inj1,type,injury])
belief(A,goal(M,belief(A,[inj1,type,injury])))
belief(A,belief(M,[inj1,type,injury]))
belief(A,inform(M,A,[inj1,agent,p1]))
belief(A,[inj1,agent,p1])
belief(A,goal(M,belief(A,[inj1,agent,p1])))
belief(A,belief(M,[inj1,agent,p1]))
	(2) **Advisor: Ok . . .**
	belief(A,acknowledge(A,M,[p1,type,person]))
	belief(A,belief(M,belief(A,[p1,type,person])))
	goal(A,belief(M,belief(A,[p1,type,person])))
	
	belief(A,acknowledge(A,M,[inj1,type,injury]))
	goal(A,belief(M,belief(A,[inj1,type,injury])))
	belief(A,belief(M,belief(A,[inj1,type,injury])))
	
	belief(A,acknowledge(A,M,[inj1,agent,p1]))
	goal(A,belief(M,belief(A,[inj1,agent,p1])))
	belief(A,belief(M,belief(A,[inj1,agent,p1])))
	goal(A,[p1,status,stable])

(3) **Advisor: What type of injury?**
belief(A,question(A,M,[inj1,injury_type]))
belief(A,goal(M,belief_wh(A,[inj1,injury_type])))
goal(A,belief_wh(A,[inj1,injury_type]))
	(4) **Medic: He's bleeding.**
	belief(A,inform(M,A,[inj1,injury_type,bleeding]))
	belief(A,[inj1,injury_type,bleeding])
	belief(A,goal(M,belief(A,[inj1,injury_type,bleeding])))
	belief(A,belief(M,[inj1,injury_type,bleeding]))

(5) **Advisor: How bad is the bleeding?**
belief(A,question(A,M,[inj1,extent]))
belief(A,goal(M,belief_wh(A,[inj1,extent])))
goal(A,belief_wh(A,[inj1,extent]))
	(6) **Medic: Pretty bad, I think it's an artery.**
	belief(A,inform(M,A,[inj1,extent,artery]))
	belief(A,[inj1,extent,artery])
	belief(A,goal(M,belief(A,[inj1,extent,artery])))
	belief(A,belief(M,[inj1,extent,artery]))

(7) **Advisor: Ok. Where's the injury?**
belief(A,acknowledge(A,M,[inj1,extent,artery]))
belief(A,belief(M,belief(A,[inj1,extent,artery])))
goal(A,belief(M,belief(A,[inj1,extent,artery])))
belief(A,question(A,M,[inj1,location]))
belief(A,goal(M,belief_wh(A,[inj1,location])))
goal(A,belief_wh(A,[inj1,location]))
	(8) **Medic: It's on the left leg.**
	belief(A,inform(M,A,[inj1,location,left_leg]))
	belief(A,[inj1,location,left_leg])
	belief(A,goal(M,belief(A,[inj1,location,left_leg])))
	belief(A,belief(M,[inj1,location,left_leg]))

dialogue ends. This conversation involves the same types of speech acts, abductive reasoning, and skill execution as occurred in the previous scenario. The system repeatedly infers the elder's mental state and takes his imputed goals and beliefs into account when selecting its own speech acts, using its knowledge to respond as appropriate. However, there is no overlap in domain predicates or domain knowledge with the prior example.

The additional two domains we have used to test our system are a 911-call domain in which a person with some type of emergency gets assistance from an operator (the dialogue system) and a physical therapy scenario in which a patient at a remote location interacts with the dialogue system to carry out a physical therapy session. As in the medical and elderly assistance domains, these involve a user with limited expertise receiving help from the dialogue system to complete a task. An interesting aspect of the physical therapy scenario is that we extended our system so that it could process input from sensors in addition to dialogue input. In other words, in that scenario, the dialogue is influenced both by what the user says and by the sensor readings. Gabaldon, Meadows, and Langley (2014) present a detailed exposition of how the system works in this domain.

To further demonstrate generality, we have run the architecture on five sequences of speech acts for the medic domain and two for the elder assistance domain. We held the domain-level knowledge constant within each domain and used the same meta-level rules for all of the runs. In each case, the system operated successfully over the input files that contained the partner's speech acts. We obtained similar results on another two domains. Although additional runs would offer further support for our claims of generality, the existing results provide encouraging evidence that the architecture can interleave domain-level and meta-level knowledge to carry out extended interactions, and that our meta-level rules for dialogue generalize across domain content. In summary, the results suggest that our framework is viable and that it deserves further study and elaboration.

We also maintain that our approach to reasoning about other agents' mental states is not limited to dialogue. Similar meta-level rules, combined with our mechanisms for abductive inference, goal generation, and skill execution, should apply equally well to other situations that involve social interaction among goal-directed agents. Scenarios that involve cooperation and competition are prime candidates for computational studies of social cognition, as are reasoning about emotions and moral judgments. Whether our new architecture can operate successfully in these arenas remains an open question, but our initial results give cause for optimism.

5. Relation to Prior Research

As noted earlier, our research makes three main contributions. First, *it provides a representational framework that cleanly separates domain-level from meta-level content*. Second, *it offers an architecture that integrates conceptual inference for situation interpretation with skill execution for goal achievement*, with these processes operating at both levels. Third, *it demonstrates both ideas in the context of a high-level dialogue system that operates across multiple domain predicates*. Although none of these is entirely novel by itself, we maintain that their combination constitutes an important contribution to the field of cognitive systems. However, for the sake of completeness, we should review previous work on each of these topics.

Nearly all artificial intelligence research distinguishes between domain-specific content and strategies for operating over it, but the vast majority encodes strategies in procedural form. Our framework differs by representing the latter as domain-independent, meta-level rules. Although this approach is seldom practiced, the idea of meta-level knowledge has a long history in artificial intelligence. Two well-known cognitive architectures – Soar (Laird, Newell, & Rosenbloom, 1987) and Prodigy (Carbonell, Knoblock, & Minton, 1990) – incorporate meta-level rules to guide problem solving. Papers on these systems emphasize domain-specific content, but both frameworks support domain-independent rules. In addition, logical analyses of action (Gelfond & Lifschitz, 1998; Reiter, 2001) often include meta-level knowledge about causal influences, and work in the BDI tradition (Rao & Georgeff, 1991) also incorporates this idea. Nevertheless, this approach remains rare in cognitive systems research and it deserves far more attention.

Most AI work focuses on components of intelligent behavior, whereas our research offers an integrated architecture for cognition that combines conceptual inference for interpretation with skill execution for goal-directed activity. Again, this is a well-established idea that has been explored in the context of both cognitive architectures and robotic architectures. However, most work on these topics has emphasized either understanding-oriented inference (e.g., Cassimatis, 2006) or activity-oriented execution (e.g., Anderson & Lebiere, 1998). On this dimension, our framework comes closest to ICARUS (Langley, Choi, & Rogers, 2009), which has modules for both conceptual inference and skill execution. But again, previous work in these paradigms has focused on interpreting and executing domain knowledge rather than meta-level content. An important exception is research on *meta-cognition* (Cox, 2005), in which systems like Meta-Acqua (Cox, 2007) and Augur (Jones & Goel, 2012) incorporate domain-independent rules to monitor, detect errors, and learn over domain-level activities. Our approach has much in common with this tradition, although our mechanisms interleave meta-level and domain content, rather than using the former to inspect traces of the latter.

Dialogue has received increased attention in recent years, but much of the research has emphasized spoken-language systems. These raise enough challenges at the speech level that most efforts utilize relatively simple dialogue models. One of the most advanced dialogue managers, RavenClaw (Bohus & Rudnicky, 2009), separates some domain-independent aspects of dialogue management from the domain level, but the domain-independent components are mainly about dialogue management procedures such as turn taking, timing, and error handling. In contrast, our aim is to take advantage of abstraction at the level of dialogue knowledge. This includes extracting domain-independent principles about whether a sequence of utterances form a valid dialogue independently of their domain-specific content and expressing them in a manner that can be used for both interpretation and generation. Another difference is that RavenClaw emphasizes dialogue generation, while our architecture supports dialogue interpretation by creating models of participants' mental states. The separation of meta-level knowledge about dialogue is made possible by our use of meta-level relations about beliefs, goals, speech acts, and dialogue structures.

Another related system, Collagen (Rich, Sidner, & Lesh, 2001), shares our concern with hierarchical plan structures for dialogue generation and also constructs models of participating agents' beliefs for use during interpretation and generation. One key difference from our system is that, except for a set of rules that describe speech acts similar to those used here, Collagen does not separate meta-level from domain knowledge. Another difference is that the earlier system encodes knowledge in procedural rather than declarative terms. Our framework also draws upon a generalized mechanism for abductive inference that operates over its knowledge structures, so that the two systems' operating details are quite different, even though they share some high-level assumptions.

We should also mention TRIPS (Ferguson & Allen, 1998), an integrated system that carries out dialogues with users to help them generate plans. This incorporates ideas from speech act theory, uses knowledge and the current plan to interpret input from the user, and generates utterances in response. One important difference is that TRIPS was designed for the task of plan creation, while our architecture is generic enough to support other collaborative tasks, provided it is given sufficient domain-specific knowledge. Moreover, because we have focused on aiding novices, our dialogues revolve around the system guiding the user in carrying out some hierarchical activity. In contrast, TRIPS utterances often involve confirming or rejecting instructions from the user for altering a plan, which requires less inference of the user's beliefs and goals.

A more recent system, FORRSooth (Epstein et al., 2012), uses a modular set of advisors to solve the problem of *grounding* in the context of requesting books from a library. Although this system is centrally concerned with inferring a user's intent in order to help him or her achieve a goal, the tasks lack the hierarchical structure that has been our focus, and system utterances mainly involve recommendations and clarification requests. However, FORRSooth's advisors operate at different levels of abstraction, with some encoding high-level strategies for making dialogues more effective, while other content is library specific. Thus, although the two systems address quite different tasks, they share the distinction between domain-level and dialogue-level knowledge, as well as a commitment to combining them dynamically during joint activities.

Finally, we should mention a much older, connected body of research from the 1980s and 1990s that also distinguished these two forms knowledge, and that has influenced our thinking about both dialogue and social cognition. In particular, our rules for speech acts borrow from Allen and Perrault's (1980) logical analysis of this topic. Litman's (1985) work comes the closest to our abductive approach to dialogue understanding, although her system also incorporated linguistic cues that ours does not. Both Bullwinkle (1975) and Hobbs et al. (1993) have emphasized the role of abduction in language processing, but they focused on sentence interpretation rather than the dialogue level.

Carberry and Lambert (1999) carry out a more detailed analysis of understanding subdialogues, which we have not emphasized, whereas McRoy and Hirst (1995) explicitly utilized an abduction mechanism to infer, and support repair of, misunderstood speech acts. Josyula et al. (2007) have addressed similar issues in their work on metacogntion. Their Alfred system operates as an interface between humans and systems in task-oriented domains, where it detects and recovers from miscommunications by engaging in a dialogue that refers to previous parts of the dialogue itself. We acknowledge our intellectual debt to these early efforts, but we have also moved beyond them to incorporate their ideas about speech acts, mental states, and abductive inference in an integrated architecture that carries out complete dialogues about complex joint activities.

6. Discussion

We have claimed that it is not only possible but advantageous for a cognitive architecture to store and utilize abstract knowledge that is useful across different domains. However, instantiating this idea effectively requires careful choices about how to represent such meta-level content. In our new architecture, these choices include:

- explicit structures, stored in the working memory, for the mental states of interacting agents in terms of meta-level predicates for beliefs and goals;
- information about the meaning of communicative events in terms of meta-level predicates for different types of speech acts;
- rules that specify the conditions and effects of speech acts in terms of the beliefs and goals of the communicating agents; and
- abstract grammatical knowledge about what sequences of speech acts constitute valid dialogues between interacting agents.

We have implemented these tenets in an integrated architecture and demonstrated its effectiveness in two domains, using the same meta-level knowledge in each case but using completely different domain-specific knowledge and predicates. None of the above ingredients are novel in isolation. Earlier systems have incorporated the notion of speech acts and even encoded their effects in terms of meta-level rules. However, we believe that our architecture is the first to combine these different types of knowledge in an integrated cognitive system that supports both the interpretation and generation of extended dialogue.

We have already discussed in some detail previous work on related topics, but it is worth mentioning again, in light of the above comments, that few dialogue systems partition knowledge into the two levels and use them for dialogue interpretation and generation. This should not be too surprising, since systems like RavenClaw, Collagen, and FORRSooth were developed for a reasonably narrow set of tasks, rather than coming from the tradition of cognitive architectures. This does not detract from their many contributions, but it does mean that we have developed a novel system that provides insights into the integrated use of different forms of knowledge.

Despite these encouraging results, there remain many open issues that we should address in future research. Our analysis of dialogue processing, even at the high level, is less complete than some earlier treatments. A fuller analysis would utilize a finer-grained taxonomy of speech acts, add explicit support for subdialogues (Carberry & Lambert, 1999), and incorporate methods for recovering from misunderstandings (McRoy & Hirst, 1995). Moreover, the architecture itself remains in the preliminary stages. The abductive inference module is unable to correct faulty assumptions through belief revision, skill execution lacks the ability to carry out activities in parallel, and the framework does not yet incorporate problem-solving methods to handle unfamiliar problems at either the domain or dialogue levels.

Some of the earlier work on dialogue processing also addresses discourse structure (Polanyi & Scha, 1984; Grosz & Sidner, 1986; Kamp & Reyle, 1993; Traum & Allen, 1994; Asher & Lascarides, 2003). In this work, researchers have devised representations of how utterances connect to form coherent discourses and have developed mechanisms that work on these representations to solve linguistic problems like anaphora resolution. Our dialogue grammars specify similar discourse level structures on top of speech act encodings of dialogue elements, but we have abstracted away from issues like anaphora resolution. This has let us use relatively simple logical representations for utterances and dialogue structure. However, we will need to elaborate these representations and deal with anaphora and other linguistic problems when we attempt to automate the translation of utterances into logical form.

Finally, we have argued that, although we have demonstrated the new architecture's operation on task-oriented dialogues, its representations and mechanisms should carry over to other settings that involve social cognition. Thus, we should test the framework in cooperative scenarios that involve agents who help each other without verbal communication, as well as competitive settings in which they have differing goals. The latter in particular will let us model scenarios in which issues like ignorance and deception (e.g., Bridewell & Isaac, 2011) influence social interactions. We predict that meta-level knowledge and reasoning play key roles in such cases, but that remains an empirical question to be studied in future research.

7. Concluding Remarks

In this paper, we presented an integrated architecture that encodes, and reasons about, the beliefs and goals of other agents in the context of task-oriented dialogues. We described the framework's memories and their contents, from domain-level and meta-level knowledge about concepts and skills to short-term elements that refer to environmental situations, physical activities, and speech acts. We also explained the architecture's mechanisms for abductive inference, goal generation, and reactive execution, which operate at both levels of abstraction.

One novel aspect of our research lies in its integration of abductive interpretation for dialogue understanding with goal-directed skill execution for dialogue generation. Another innovation is its use of both domain-level and meta-level knowledge for these distinct but complementary cognitive tasks within a coherent architecture. To a large degree, this is facilitated by an explicit representation of agents' beliefs and goals and by assigning meaning to dialogues in these terms. Using this architecture, we developed systems that carry out, at the level of speech acts, extended dialogues about goal-directed activities. We also demonstrated that the same meta-level knowledge about dialogue operates with different domain content, suggesting that the approach has considerable generality.

Acknowledgments

This research was supported in part by Grants N00014-09-1-1029 and N00014-10-1-0487 from the Office of Naval Research. We thank Chitta Baral, Paul Bello, Will Bridewell, Herb Clark, Tolga Könik, David Stracuzzi, and Richard Weyrauch for discussions that influenced the approach we have reported here.

References

Allen, J., & Perrault, C. (1980). Analyzing intention in utterances. *Artificial intelligence*, *15*, 143–178.

Anderson, J., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum.

Asher, N., & Lascarides, A. (2003). *Logics of conversation*. Cambridge University Press.

Austin, J. L. (1962). *How to do things with words*. Cambridge, MA: Harvard University Press.

Bohus, D., & Rudnicky, A. (2009). The RavenClaw dialog management framework: Architecture and systems. *Computer Speech & Language*, *23*, 332–361.

Bridewell, W., & Isaac, A. (2011). Recognizing deception: A model of dynamic belief attribution. *Proceedings of the AAAI 2011 Fall Symposium on Advances in Cognitive Systems* (pp. 50–57). Arlington, VA: AAAI Press.

Bullwinkle, C. L. (1975). Picnics, kittens and wigs: Using scenarios for the sentence completion task. *Proceedings of the Fourth International Joint Conference on Artificial Intelligence* (pp. 383–386). Tbilisi, USSR.

Carberry, S., & Lambert, L. (1999). A process model for recognizing communicative acts and modeling negotiation subdialogues. *Computational Linguistics*, *25*, 1–53.

Carbonell, J., Knoblock, C., & Minton, S. (1990). Prodigy: An integrated architecture for planning and learning. In K. Van Lehn (Ed.), *Architectures for intelligence*. Hillsdale, NJ: Lawrence Erlbaum.

Cassimatis, N. (2006). A cognitive substrate for achieving human-level intelligence. *AI Magazine*, *27*, 45.

Choi, D. (2010). *Coordinated execution and goal management in a reactive cognitive architecture*. Doctoral dissertation, Department of Aeronautics and Astronautics, Stanford University, Stanford, CA.

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

Cox, M. (2007). Perpetual self-aware cognitive agents. *AI Magazine*, *28*, 32–45.

Cox, M. T. (2005). Metacognition in computation: A selected research review. *Artificial Intelligence*, *169*, 104–141.

Epstein, S. L., Passonneau, R. J., Gordon, J., & Ligorio, T. (2012). The role of knowledge and certainty in understanding for dialogue. *Cognitive Systems*, *1*, 93–108.

Ferguson, G., & Allen, J. F. (1998). TRIPS: An integrated intelligent problem-solving assistant. *Proceedings of the Fifteenth National Conference on Artificial Intelligence* (pp. 567–572).

Gabaldon, A., Meadows, B., & Langley, P. (2014). Integrated interpretation and generation of task-oriented dialog. *Proceedings of the Fifth International Workshop on Spoken Dialog Systems*. Napa, CA.

Gelfond, M., & Lifschitz, V. (1998). Action languages. *Electronic Transactions on Artificial Intelligence*, *3*, 195–210.

Grosz, B. J., & Sidner, C. L. (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics*, *12*, 175–204.

Hobbs, J. R., Stickel, M. E., Appelt, D. E., & Martin, P. A. (1993). Interpretation as abduction. *Artificial Intelligence*, *63*, 69–142.

Jones, J. K., & Goel, A. K. (2012). Conceptual semantics of domain knowledge in learning by correcting mistakes. *Poster Collection: The First Annual Conference on Advances in Cognitive Systems* (pp. 53–68).

Josyula, D. P., Fults, S., Anderson, M. L., Wilson, S., & Perlis, D. (2007). Application of MCL in a dialog agent. *Proceedings of the Third Language and Technology Conference*. Poznan, Poland.

Kamp, H., & Reyle, U. (1993). *From discourse to logic*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, *33*, 1–64.

Langley, P. (2012). The cognitive systems paradigm. *Advances in Cognitive Systems*, *1*, 3–13.

Langley, P., Choi, D., & Rogers, S. (2009). Acquisition of hierarchical reactive skills in a unified cognitive architecture. *Cognitive Systems Research*, *10*, 316–332.

Litman, D. (1985). *Plan recognition and discourse analysis: An integrated approach for understanding dialogues*. Doctoral dissertation, Department of Computer Science, University of Rochester, Rochester, NY.

McRoy, S., & Hirst, G. (1995). The repair of speech act misunderstandings by abductive inference. *Computational Linguistics*, *21*, 435–478.

Polanyi, L., & Scha, R. (1984). A syntactic approach to discourse semantics. *Proceedings of the Tenth International Conference on Computational Linguistics* (pp. 413–419). Stanford, CA.

Rao, A. S., & Georgeff, M. P. (1991). Modeling rational agents within a BDI-architecture. *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning* (pp. 473–484). Cambridge, MA: Morgan Kaufmann.

Reiter, R. (2001). *Knowledge in action: Logical foundations for describing and implementing dynamical systems*. Cambridge, MA: MIT Press.

Rich, C., Sidner, C. L., & Lesh, N. (2001). Collagen: Applying collaborative discourse theory to human-computer interaction. *AI Magazine*, 15–25.

Searle, J. (1969). *Speech acts: An essay in the philosophy of language*. New York: Cambridge University Press.

Traum, D. R., & Allen, J. F. (1994). Discourse obligations in dialogue processing. *Proceedings of the Thirty-Second Annual Meeting on Computational Linguistics* (pp. 1–8). Stroudsburg, PA.