Integrated Cognition
AAAI Technical Report FS-13-03

Meta-Level and Domain-Level Processing in Task-Oriented Dialogue

Alfredo Gabaldon
Silicon Valley Campus
Carnegie Mellon University
Moffett Field, CA 94035 USA
alfredo.gabaldon@sv.cmu.edu

Pat Langley
Silicon Valley Campus
Carnegie Mellon University
Moffett Field, CA 94035 USA
patrick.w.langley@gmail.com

Ben Meadows
Computer Science Department
University of Auckland
Private Bag 92019, Auckland, NZ
bmea011@auckland.ac.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. We present an integrated architecture that incorporates this distinction and that uses it both during conceptual inference to interpret its situation and during skill execution to achieve its goals.

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 and carry out 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.

Our main claim is 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 separates domain-specific from general knowledge, including earlier efforts on dialogue processing. We will not claim that we are the first to separate domain-level from meta-level knowledge, even in the area of dialogue, but we will argue that this approach holds great promise and that it deserves more attention within the cognitive systems community.

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 (the caller) is confronting a medical emergency and seeks another's advice (the advisor). The latter has the expertise needed to solve the problem, but cannot affect the environment directly, whereas the caller lacks expertise but can act if provided with appropriate instruction.

Caller:	I have a medical emergency!
Advisor:	Do you need help or someone else?
Caller:	It's my father.
Advisor:	Okay. Tell me what happened.
Caller:	He just fainted.
Advisor:	I understand. Is he conscious now?
Caller:	No, he's not responding.
Advisor:	We need to check if he is breathing.
Caller:	Okay. What should I do?
Advisor:	Check for breathing sounds by placing your ear near his nose.
Caller:	Okay ... I don't hear anything.
Advisor:	Do you have a mirror in your purse?
Caller:	Yes
Advisor:	Hold the mirror under his nose, see if it fogs.
Caller:	Okay ... Yes, the mirror is fogging.
Advisor:	Good. Now, raise his legs above head level.
Caller:	Okay, done.
Advisor:	Now wait, the ambulance will arrive soon.
Caller:	Okay. Thank you.

This idealized 911 call, despite its simplicity, raises many of the issues that we want to address. Here are some noncontroversial observations about 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;
- Participants develop a shared model not only of the environmental situation, but of each others' beliefs and goals; this joint model 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 their 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; carry out goal-directed activities appropriate to this inferred model; and utilize both 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 an integrated architecture that responds to these requirements. Although we illustrate its 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, 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, and Rogers 2009), a cognitive architecture designed for physical agents. In this section, we describe the new framework's memories 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.

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 *belief*(A, C) and *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 a communicative act.

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 *knows_if(A, C)* and *not_knows_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 relation R that occurs in *triple*(R, X, Y)), we use *knows_wh*(A, R, X) and *not_knows_wh*(A, R, X) to denote that agent A *knows what* and does *not know what*, respectively, a corresponding Y for the relation R. For instance, *belief*$(A,$ *knows_wh*$(B, type, event1))$ represents A's belief that B knows event1's type is emergency. 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.

Domain-level literals encode information related to the domain. These always occur as arguments of some meta-level predicate; 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 *belief(Advisor, holding(caller, mirror))*, it would store the three equivalent elements, *belief(Advisor, triple*$(type, event1, holding))$, *belief(Advisor, triple*$(agent, event1, caller))$, and *belief(Advisor, triple*$(object, event1, mirror))$. Note that this encoding also reifies the event or relation.

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 seven types that appear sufficient for our purposes:

- *inform*(S, L, C): speaker S asks L to believe content C;
- *acknowledge*(S, L, C): S tells L it has received and now believes content C;
- *question_wh*(S, L, C): S asks L a 'wh' question C;
- *question_if*(S, L, C): S asks L an 'if' question C;
- *propose*(S, L, C): S asks L to adopt goal C;
- *accept*(S, L, C): S tells L it has adopted goal C;
- *reject*(S, L, C): S tells L it has rejected goal C.

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

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 supporting inference and skills enabling goal-directed activity. We can partition this into domain-level and meta-level knowledge.

Domain-level concepts are specified by rules in which the head comprises the conceptual predicate and its arguments, and the body describes a relational pattern which that concept denotes. 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 takes the form of a hierarchy, with more abstract 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, along with a body that includes conditions that must be satisfied for it to execute and its expected effects. The bodies of primitive skills refer to executable actions, whereas nonprimitive skills include a sequence of lower-level skills. For example, the nonprimitive skill

$$\begin{aligned} &skill(handle\_medical\_emergency(EM)) \leftarrow \\ &\qquad triple(type, EM, medical), \\ &\qquad skill(ask\_wh\_question(patient, EM)), \\ &\qquad skill(determine\_if\_conscious). \end{aligned}$$

states that, to handle a medical emergency, one should first ask about the identify of the patient and then determine if he or she is conscious. Different rules with the same head denote alternative ways to decompose a skill. Thus, as in the ICARUS architecture, the set of skills is equivalent to a hierarchical task network.

A third type of domain-level knowledge consists of **goal-generating rules**. These differ from concepts and skills in that they do not define new predicates, but rather describe the conditions under which the agent 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, causes the agent to establish a goal of stabilizing that person. Choi (Choi 2010) has reported an extension of ICARUS that uses similar knowledge, although his approach proposed goals about desired states, whereas ours generates desired activities.

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 the different **speech act rules**. For example, the conceptual rule for *propose*, from the speaker's perspective, can be stated:

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

which 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 S has adopted C as a goal.[1] 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 for the listener.

Another form of meta-level knowledge is a **dialogue grammar** that specifies relations among speech acts. This includes recursive rules: for instance, a dialogue may consist of S asking L a question Q, followed by S answering L with A, followed by another dialogue. 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 speech acts, along with 13 meta-level rules that determine whether they are satisfied. For example, the rules for *wh* questions ensure that answers are consistent with the agent's beliefs. As with speech act rules, those for the dialogue grammar make no reference to any domain predicates.

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

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 dealt only with physical environments, it could rely on deduction to draw conclusions about its situation. However, social cognition, including dialogue, requires making plausible assumptions about unobserved mental states.

For this reason, the architecture utilizes an abductive inference module 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.[2] The abductive mechanism prefers the

[1] A more complete rule would include temporal constraints which specify that some relations hold before the speech act and that others hold afterward.

[2] The current implementation does not yet support belief revision, which is clearly needed in cases of misunderstanding.

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 it exceeds a limit. Once the architecture finds an explanation, it adds the default assumptions to working memory, making these elements givens on the next round, so that typically it must introduce only a few new assumptions per cycle.

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 is able to introduce 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 acknowledgements, 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 not a desired state, but rather a desire to carry out some action or activity. In addition, abductive inference may introduce new top-level goals, as when the agent adopts an objective that another has proposed.

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 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 down through the skill hierarchy that is relevant to achieving the top-level goal. Upon reaching a primitive skill, the architecture carries out its associated actions after replacing variables with their bindings.

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 have adopted a new goal such as $goal(caller, \textit{belief}(advisor, breathing(patient)))$, 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. For instance, 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 of ICARUS, as seems appropriate for an architecture that carries out activities over time.

4 Generality of the Approach

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 focus on two domains, one involving 911 calls like the one discussed earlier and the other involving a scenario in which the system[3] helps an elderly person track her medications.

Our experimental protocol provides the system with access to a text file that contains a sequence of speech acts for the person who needs assistance. 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 911 call scenario, the system acts as an agent, the $advisor$, who is helping another agent, the $caller$, deal with a medical emergency situation. The domain is described using a dozen conceptual predicates like *breathing* and *emergency*. The domain knowledge includes one goal-generating rule (to handle any reported emergency) and two conceptual rules. The latter two rules infer that a patient is breathing

[3] By 'system', we mean the architecture plus the meta-level and domain-level knowledge that we provide it.

given that a mirror fogs when placed under his nose or that certain sounds are heard. The domain knowledge also includes 15 skills, including:

$$\begin{aligned}&\textit{skill}(\textit{determine\_if\_breathing}(Patient)) \leftarrow\\&\quad\quad \textit{not\_knows\_if}(Listener, \textit{breathing}(Patient)))),\\&\quad\quad \textit{skill}(\textit{ask\_if\_question}(Listener, \textit{has\_mirror}))),\\&\quad\quad \textit{skill}(\textit{check\_breathing\_with\_mirror}(Patient)).\end{aligned}$$

This skill encodes one way of finding out if a patient is breathing. The condition states that the speaker believes the listener does not know if the patient is breathing. If the condition is satisfied, the skill invokes two subskills: one that asks the listener it she is in possession of a mirror (see dialogue in Sec. 2) and another that invokes a skill to check for breathing using the mirror. This example is interesting because a domain skill uses a meta-level skill to ask a question, 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 caller has an emergency, the system applies abductive inference and updates its working memory by adding a belief that the caller has an emergency, that she has a goal for the advisor to believe this, and a belief that she believes the advisor now believes she has an emergency. The system also generates and adds to working memory the top-level goal of handling the emergency. Next, the skill execution process responds with an acknowledge speech act, after which it invokes a high-level skill for handling emergencies. This leads it to produce a question speech act that asks the caller to state the type of emergency.

The dialogue then continues with an exchange of similar questions and answers about the patient and the nature of the problem. From the caller's answers, the system infers that she does not know if the patient is breathing, which triggers execution of skills to find out. After further questions and instructions to the caller that, eventually, determine that she believes the patient is breathing, the advisor issues a final instruction to wait for the ambulance and ends the interaction. Throughout the conversation, the system uses its dialogue grammar to update its beliefs and goals, and it draws on its 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 caller answers the advisor's early question about breathing in the affirmative, then the system does not execute its skills for addressing it, changing the path of the conversation.

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 predicates. Here we provide the system with one goal-generating rule related to getting the elder to take his meds, along with 12 domain-level skills. In this scenario, the system initially believes that it is time for elder to take his medicine, so the goal-generation process immediately adds a top-level goal to this end, which in turns leads to execution of a relevant skill. The dialogue starts with the system taking the initiative to inform the elder it is time to take his meds, which the elder acknowledges, after which it informs him 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 Elder informs the system that he left the pills in the living room, so it proposes that he get them. Finally, the system reminds elder to take his pills, the elder responds that he has done so, and the dialogue ends. At an abstract level, 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. However, there is no overlap in domain predicates or domain knowledge with the prior example.

To further demonstrate generality, we have run the architecture on two sequences of speech acts for both the 911 and elder assistance domains. 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. 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 worth 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 judgements. Whether our new architecture can operate successfully in these arenas remains an open question, but our initial results give cause for optimism.

5 Related Research

As noted earlier, our work 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 themselves, together they constitute an innovative contribution to cognitive systems. However, for the sake of completeness, we should review related work on each of these topics.

Nearly all AI 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 AI. Two well-known cognitive architectures, Soar (Laird, Newell, and Rosenbloom 1987) and Prodigy (Carbonell, Knoblock, and Minton 1990), incorporate meta-level rules to guide problem solving; these typically encode domain-specific content, but both frameworks also support domain-independent rules. Logical analyses of action (e.g. (Gelfond and Lifschitz 1998; Reiter 2001)) often include meta-level knowledge about causal influences, and work in the BDI tradition (Rao and Georgeff 1991) also incorporate 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 and Lebiere 1998)). On this dimension, our framework comes closest to ICARUS (Langley, Choi, and Rogers 2009), which has modules for both conceptual inference and skill execution. But again, previous work in these traditions has focused on interpretation and execution of domain-level knowledge, with little if any content being stated at the meta-level. One important exception is Meta-Aqua (Cox 2007), which incorporates domain-independent rules to support meta-cognitive detection of errors and to drive learning.

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 and Rudnicky 2009), encodes strategies in procedural terms. Another recent system, FORRSooth (Epstein et al. 2012), uses a modular set of advisors, to guide dialogue, but these do not appear to be stated in terms of meta-level knowledge. In contrast, a coherent body of research in the 1980s and 1990s (e.g., (Allen and Perrault 1980; Carberry and Lambert 1999; Litman 1985; McRoy and Hirst 1995)) explicitly distinguished domain-level from dialogue-level knowledge. This work, including its analysis of speech acts, has strongly influenced our own approach, which differs mainly in embedding these ideas in an integrated architecture and in its emphasis on complete dialogues.

6 Concluding Remarks

In this paper, we presented a new architecture that encodes, and reasons about, the beliefs and goals of other agents using a mixture of domain-level and meta-level knowledge. We described the architecture in terms of its memories and their contents, along with mechanisms for abductive inference, goal generation, and skill execution. Using this architecture, we developed systems that carry out, at the level of speech acts, extended dialogues about goal-directed activities. We demonstrated that the same meta-level knowledge about dialogue operates with different domain content, suggesting that the approach has considerable generality.

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 and Lambert 1999), and incorporate methods for recovering from misunderstandings (McRoy and Hirst 1995). Moreover, the architecture itself remains in the preliminary stages. The abductive inference module is unable to revise faulty assumptions, skill execution lacks the ability to carry out skills in parallel, and the framework does not yet incorporate a problem-solving module to handle unfamiliar problems.

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 and 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 for future research.

Acknowledgements

This research was supported in part by Grant N00014-09-1-1029 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 to we have reported here.

References

Allen, J., and Perrault, C. 1980. Analyzing intention in utterances. *Artificial intelligence* 15(3):143–178.

Anderson, J., and Lebiere, C. 1998. *The atomic components of thought*. Lawrence Erlbaum.

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

Bohus, D., and Rudnicky, A. 2009. The RavenClaw dialog management framework: Architecture and systems. *Computer Speech & Language* 23(3):332–361.

Bridewell, W., and Isaac, A. 2011. Recognizing deception: A model of dynamic belief attribution. In *Advances in Cognitive Systems: Papers from the 2011 AAAI Fall Symposium*, 50–57.

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

Carbonell, J.; Knoblock, C.; and Minton, S. 1990. Prodigy: An integrated architecture for planning and learning. In Lehn, K. V., ed., *Architectures for intelligence*. Hillsdale, NJ: Lawrence Erlbaum.

Cassimatis, N. 2006. A cognitive substrate for achieving human-level intelligence. *AI Magazine* 27(2):45.

Choi, D. 2010. *Coordinated execution and goal management in a reactive cognitive architecture*. Ph.D. Dissertation, Stanford University.

Cox, M. 2007. Perpetual self-aware cognitive agents. *AI magazine* 28(1):32–45.

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

Gelfond, M., and Lifschitz, V. 1998. Action languages. *Electronic Transactions on Artificial Intelligence* 3:195–210.

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

Langley, P.; Choi, D.; and Rogers, S. 2009. Acquisition of hierarchical reactive skills in a unified cognitive architecture. *Cognitive Systems Research* 10(4):316–332.

Litman, D. 1985. Plan recognition and discourse analysis: An integrated approach for understanding dialogues. PhD Thesis, Technical Report 170, Department of Computer Science, Univ. of Rochester.

McRoy, S., and Hirst, G. 1995. The repair of speech act misunderstandings by abductive inference. *Computational linguistics* 21(4):435–478.

Rao, A. S., and Georgeff, M. P. 1991. Modeling rational agents within a BDI-architecture. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, 473–484.

Reiter, R. 2001. *Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems*. Cambridge, MA: MIT Press.

Searle, J. 1969. *Speech Acts: An Essay in the Philosophy of Language*. New York, NY: Cambridge University Press.