Talking to Computers*

Michael L. Anderson[1], **Darsana P. Josyula**[2], **Don Perlis**[1,2]
(1)Institute for Advanced Computer Studies,
(2)Department of Computer Science,
University of Maryland, College Park, MD 20742
{anderson,darsana,perlis}@cs.umd.edu

Abstract

Our broad claim is that time-sensitive metareasoning can enhance the ability of natural language HCI systems to converse with human interlocutors, by giving these systems both the time-awareness and meta-linguistic skills (including especially the ability to recognize and repair dialog problems, by learning if need be) which appear to be necessary for free, flexible, and natural conversation. We illustrate this enhancement with a description of our ongoing work in cooperative natural language HCI systems.

1 The Nature of Conversation

The development of a truly viable natural language computer interface will require advances in computer reasoning, in particular in the ability to reason about and manage the particular uncertainties inherent in interactive dialog. Natural language alone is complex and ambiguous, and dialog is even more so, for there is significance not just in the words themselves, but also in such things as tone, emphasis, gesture, and timing, each of which contributes to the meaning of an utterance, and which should therefore be considered important, interacting parts of the whole. Although all these elements generally work together to *aid* comprehension (the command "Come here!" is delivered with a characteristic tone and accompanied by an appropriate waving gesture) still it is possible to get conflicting signals, or to make mistakes in one element of an interpretation which can generate apparent conflicts among the various elements of an utterance (if you heard "Get clear!", the tone would be appropriate, but you might well wonder why the speaker was waving you towards him). Thus it is important to avoid simply allowing the weighted sum of these various elements to determine an interpretation; rather, one should recognize and make judgments about anomolies as they arise, using such anomolies as opportunities for correcting mistakes, or at least confirming one's favored interpretation.

This, in fact, is what human dialog partners do. They manage the uncertainty inherent in dialog by continually monitoring their conversations, their own comprehension, and the apparent comprehension of their interlocutor. This ability is apparently very basic, and fundamental to language use. Clark [Clark and Schaefer, 1989a] presents empirical evidence for the use of metalinguistic skills by young children. Among many other things, these skills include:

1 **Monitoring one's ongoing utterance**: An example of this was seen in a 2 year, 6 month child practicing parts of speech (in this case its pronunciation of "berries") on its own: "Back please/berries/*not* barries/barries, barries/*not* barries/berries /ba ba"

2 **Checking the result of an utterance**: Children at least as young as 5 years, 4 months comment on and correct the utterances of others. They also verify that the listener has understood their utterance and attempt a repair otherwise.

3 **Deliberately trying to learn**: For instance, a 4 year old will ask things like: "Mommy, is it AN A-dult or A NUH-dult?"

4 **Predicting the consequences of using inflections, words, phrases or sentences**: This includes judging the politeness of utterances, which is exhibited by children aged four and a half. Children can also correct word order in sentences judged "silly". Clark cites instances of this being done by two-year olds.

These behaviors are not just necessary to children learning a language, but in fact pervade conversation between fully competent language users [Purver, 2002; Purver *et al.*, 2002; Anderson *et al.*, forthcoming]. Dialog partners routinely elicit and provide feedback as the conversation continues, and make conversational adjustments as necessary, by employing a set of "grounding" behaviors [Clark and Schaefer, 1987; 1989b; Brennan, 1998; 2000; Brennan and Hulteen, 1995; Cahn and Brennan, 1999; Clark and Brennan, 1991; Krahmer *et al.*, 1999a; 1999b; Paek and Horvitz, 1999; Traum, 1994]. The behaviors employed to establish,

---

*This research is supported in part by the AFOSR and ONR

mantain, or confirm grounding include self-monitoring behaviors such as judging one's level of confidence in a given interpretation or comparing current statements with the implications of past ones, as well as asking questions designed to confirm another's comprehension ("Got it?") or to establish one's own ("Wait. I don't think I understand the concept of hidden variables."). This ability to ask questions about what's been said and understood—that is, the ability to engage in meta-dialog (dialog about the dialoge or its elements)—and to use the results of these meta-dialogic interchanges to help understand otherwise problematic utterances, is essential to conversation.[1] We believe that one of the problems facing natural language HCI at the present time is that the computer partners in dialog are not generally equipped with this ability to elicit and utilize appropriate feedback; this can make these systems difficult and frustrating to use [Bohlin *et al.*, 1999].

Thus, we have been inching closer (we hope) to a viable natural language computer interface by focusing our efforts on understanding the importance, role, and extent of human dialog monitoring and repair—as well as the meta-reasoning and meta-lingusitic skills this monitoring and repair involves—and implementing these behaviors in a task-oriented natural language computer interface [Perlis *et al.*, 1998; Traum *et al.*, 2002; Anderson *et al.*, 2002; forthcoming; Josyula *et al.*, 2003; Andersen *et al.*, 1999]. The specific goal we have set for ourselves is the design of a system modeled not on conversation with a fluent colleague, but rather, for example, on a task-oriented interaction with a stranger who doesn't speak much of a common language. In these situations, e.g., buying a train ticket in a foreign country, speakers are often able to communicate to effectively solve a joint task, in spite of problems in word recognition, or use of unfamiliar words or syntactic structures. Despite the difficulties of understanding the language, interactive dialog behaviors and ongoing repairs allow humans to overcome some of these problems. In what follows, we will detail our approach, and describe some of the behaviors we have been able to generate in our natural language HCI systems.

2 Active Logic and Conversation

Insofar as the above analysis of the nature of conversation is correct, it suggests that any agent hoping to participate fully in even rudimentary conversation should be:

1. Self-monitoring: A dialog agent should track its own comprehension, and maintain a history of its interpretations, so as to be able not just to notice errors (such as mismatches between expectations and obervations, or between implications of past utterances and current ones) but also to be able to trace their origins and fix them.
2. Contradiction-tolerant: Having encountered problems, which are inevitable in conversation (indeed, whenever one is dealing with a complex, dynamic environment) a dialog agent should be able to gracefully handle these situations.
3. Time-sensitive: Dialog is governed by a number of time-based expectations. If someone makes a conversational overture, or asks a question, a response is expected in a certain amount of time.[2] A dialog agent should be sensitive to these demands. Likewise, it should have expectations about the time it will take its human user to respond, as deviations could indicate a problem. For instance, a long pause could signal a turn change, or that the user is confused, or even that the user is no longer engaged in the conversation.
4. Multi-modal: A dialog agent should ideally be able to monitor all the various aspects of speech, including affect, tone and gesture, as well as to match its own utterances with appropriate accompaniment. Unfortunately, monitoring, interpreting and producing these effectively and freely (without prior scripting) is currently beyond the state of the art.[3] However, the ability to attend to multiple contexts is not, and is something which ought to be included in dialog systems whenever possible. For, to understand an utterance it may not be enough to attend to the dialog context (e.g. the subject under consideration, open questions under discussion, turn) but also to the larger environment. If one says "I guess he's had enough," the "he" might equally well refer to the fellow under discussion who quit his job, or the one who just fell from his barstool.

Our approach to implementing these meta-dialogic abilities involves three conceptual planks. First, there is the issue of representation of aspects of the dialog processing. For many object-level behaviors, it is not necessary to have an explicit representation of the processes that the system performs, other than the programming or mechanism that produces the behavior at appropriate times. However, to engage in meta-behavior, such as dialog about dialog, rich representations are needed both for producing and understanding meta-utterances. In the case of meta-dialog we are influenced by work such as [Hobbs, 1985; Hwang and Schubert, 1993; Traum *et al.*, 1996; Poesio and Traum, 1997; McRoy *et al.*, 1997], that proposes detailed logical representations of a range of dialog phenomena. Second, the system must be able to effectively use such representations in inference to be able to notice interesting phenomena, such as implications of what has been done,

---

[1] Indeed, as any teacher—and student—knows, it is important to natural language comprehension more generally.

[2] Exactly how much time is context dependent. For a conversational overture or a very easy question, a near immediate response is required. If a question is perceived to be difficult, or require complex thinking, a longer delay may well be tolerated.

[3] Although some impressive work is being done, see, e.g. [Cassell *et al.*, 2000]

recognizing resulting anomalies, and deciding what can be done about it. For use in a dialog system, this reasoning cannot be off-line, but must be integrated within the normal dialog behavior of such a system. This leads to the third plank, integration of reasoning with acting and non-logical processing. Reasoning about anomalies and meta-dialog is not enough. To be effective such reasoning should be integrated with normal functioning, being able to affect object level processes.

In our work on active logic we have been exploring ways to implement real-time metareasoning for use in spoken dialog, with special attention to pragmatics. The idea behind active logics is to use an inference mechanism that takes account of the passage of time as it performs inferences. This in turn can lend it both the expressive and the inferential power to monitor its own reasoning in a real-time fashion, as that very reasoning is going on, thus watching for errors (such as mismatches between conveyed and intended meanings); noting temporal conversational cues such as pauses that may signal a turn change; and re-examining its beliefs and altering them appropriately.

3 Active Logic: An Introduction

Active logics are a family of formalisms that combine inference rules with a constantly evolving measure of time (a 'now') that itself can be referenced in those rules. An account of the basic concepts can be found in [Elgot-Drapkin and Perlis, 1990].

In active logic, aspects of the environment are represented as first order formulas in the knowledge base. Such formulas might represent perceptions of a user's utterance, observations about the state of the domain, or rules added by a system administrator. Inference rules provide the mechanism for "using" the knowledge for reasoning. One aspect of active logic especially important in the current context is its robust ability to continue to reason normally as formulas are added, changed or deleted from its knowledge base. In other words, the evolving knowledge base is naturally integrated into the ongoing reasoning processes. This makes active logic a good candidate for a reasoning agent which is expected to observe and interact in real time with a continually changing world or domain.

One of the original motivations for active logics was that of designing formalisms for reasoning about an approaching deadline; for this use it is crucial that the reasoning takes into account the ongoing passage of time as that reasoning proceeds. Such a formalism has the ability to explicitly track the individual steps of a deduction, making it a natural mechanism for reasoning about contradictions and their causes.

Each "step" in an active logic proof itself takes one active logic time-step; thus inference always moves into the future at least one step and this fact can be recorded in the logic. The KB will at all times be finite since the finitely-many inference rules can produce only finitely-many conclusions in one time-step.[4] The meaning of an inference rule such as 1 (an active logic analogue to *modus ponens*), is that if A and $A \rightarrow B$ are in KB at time (step number) `i`, then B will be added to the KB at time `i+1`.

$$(1)\quad \begin{array}{ll} \texttt{i} & : \\ \texttt{i+1} & : \end{array} \frac{\texttt{A, A} \rightarrow \texttt{B}}{\texttt{B}}$$

(In general there may be conditions that must be met before such a rule can fire—see below; but if a rule can fire, it will.) In addition to the new formula B, the KB at step `i+1` would contain all the formulas that are inherited from step `i`. By default, all beliefs from one step that are not directly contradicting are inherited to the next step. However some beliefs like the ones related to the current time are not inherited to the next step. (See below). The inheritance of formulas from one step to the next is controlled by inheritance rules. One simple version of such an "inheritance rule", which also illustrates the use of firing conditions, is shown in 2:

$$(2)\quad \begin{array}{ll} \texttt{i} & : \\ \texttt{i+1} & : \end{array} \frac{\texttt{A}}{\texttt{A}}$$

$$\texttt{[condition: } \neg\texttt{A} \notin \texttt{KB at step i and A} \neq \texttt{Now(i)]}$$

To achieve much of their reasoning, active logics employ a notion of "now" that is constantly updated by the "clock rule" shown in 3:

$$(3)\quad \begin{array}{ll} \texttt{i} & : \\ \texttt{i+1} & : \end{array} \frac{\texttt{Now(i)}}{\texttt{Now(i+1)}}$$

An active logic keeps track of the passage of time using the current value of "now", so it is important that the value of "now" from a previous step is not inherited to the next step. The firing condition in the inheritance rule in 2 would prevent `Now(i)` lingering in KB after step `i` along with the newly inferred `Now(i+1)` [5]

Theorems can be marked with their time (step-number) of being proven, i.e., the current value of "now". This step-number is itself something that further inferences can depend on, such as inferring that a given deadline is now too close to meet by means of a particular plan under refinement if its enactment is estimated to take longer than the (ever shrinking) time remaining before the deadline.

Active logic formalisms are distinct from traditional temporal logics, in that the latter characterize truth about past, present, and future as if from a timeless (or unchanging) present; that is, the inferences do not formally correspond to an increase in the value of "now". This is appropriate as long as the temporal reasoning is by one agent about another agent far removed in time,

---

[4] In ongoing work begun in [Nirkhe *et al.*, 1997] we have been exploring ways to keep the KB size not merely finite but bounded, analogous to human short-term memory.

[5] Inheritance and disinheritance are directly related to belief revision [Gärdenfors, 1988] and to the frame problem [McCarthy and Hayes, 1969; Brown, 1987]; see [Nirkhe *et al.*, 1997] for further discussion.

or if the latter agent's activity is independent of the former. But when an agent is reasoning about its own ongoing activity, or about another agent whose activity is highly interdependent, traditional "time-frozen" reasoning is at a disadvantage, and "time-tracking" active logics can bring new power and flexibility to bear.

It is the time-sensitivity of active logic inference rules that provides the chief advantage over more traditional logics. Thus, an inference rule can refer the results of all inferences *up until now*—i.e. thru step i—as it computes the subsequent results (for step $i+1$). This allows an active logic to reason, for example, about its own (past) reasoning; and in particular about what it has *not* yet concluded. Moreover, this can be performed quickly, since it involves little more than a lookup of the current knowledge base.

Rules 2 and 3 illustrate one way in which an agent clock can be updated and also how direct contradictands can be kept from lingering, while other wffs may remain in the KB to facilitate further reasoning. Note also that, although this does "dismiss" the contradictands from further inferences, the "conflict-recognition" rule to be given below in 4, ensures that a record is kept in the KB of the former presence of a contradiction. Preserving this "historical" information is important in order to attempt a more solid repair of the contradiction.

As is well known, traditional formalisms, including most modal, temporal and nonmonotonic logics, suffer from the "swamping problem" (this is related to the "omniscience" problem of traditional logics of belief: all (infinitely-many) consequences of the axioms are theorems and hence are believed). As a result, in those logics, any possible clues as to how to proceed with reasoning when a contradiction is encountered are rendered ineffective by their own negations which are also derived from the contradiction.

There have been some attempts to overcome the swamping problem, but so far only in the propositional case, and even so the essential time-dependency for real-time capabilities is still missing there.

Even though the problem of inconsistency is treated by some logics like paraconsistent logics, in reality most of the traditional logics do not note or repair inconsistencies, they just carry on with them. Nor in general do they provide for any special real-time status as needed by a real-world agent. On the other hand, active logics are intended for on-board use by an agent, not as an external specification of an agent.

In active logics, since the notion of inference is time-dependent, it follows that at any given time only those inferences that have actually been carried out so far can affect the present state of the agent's knowledge. As a result, even if directly contradictory wffs, P and ¬P, are in the agent's KB at time `t`, it need not be the case that those wffs have been used by time `t` to derive any other wff, Q. Indeed, it may be that `t` is the first moment at which both P and ¬P have simultaneously been in KB.

By endowing an active logic with a "conflict-recognition" inference rule such as that in 4, *direct* contradictions can be recognized as soon as they occur, and further reasoning can be initiated to repair the contradiction, or at least to adopt a strategy with respect to it, such as simply avoiding the use of either of the contradictands for the time being. The `Contra` predicate is a meta-predicate: it is about the course of reasoning itself (and yet is also part of that same evolving history).

$$\begin{array}{ll} \texttt{i} & : \\ \texttt{i+1} & : \end{array} \quad \frac{\texttt{P, }\neg\texttt{P}}{\texttt{Contra(i, P, }\neg\texttt{P)}} \tag{4}$$

The idea then is that, although an indirect contradiction may lurk undetected in the knowledge base, it may be sufficient for many purposes to deal only with direct contradictions. Sooner or later, if an indirect contradiction causes trouble, it may reveal itself in the form of a direct contradiction. After all, a real agent has no choice but to reason only with whatever wffs it has been able to come up with *so far*, rather than with implicit but not yet performed inferences. Moreover, since consistency (i.e., the lack of direct or indirect contradictions) is, in general, undecidable, all agents with sufficiently expressive languages will be forced to make do with a hit-or-miss approach to contradiction detection. The best that can be hoped for, then, seems to be an ability to reason effectively in the presence of contradictions, taking action with respect to them only when they become revealed in the course of inference (which itself might be directed toward finding contradictions, to be sure).

Unlike most NMR formalisms, we do not attempt to capture the (usually undecidable) absolute truth about what is consistent with what is known; this is in general impossible for real agents. If nothing is *already* known that would prevent a default conclusion, then the agent has little choice except to draw that conclusion, and this is what an active logic does. If later (with more time) the agent discovers a consequence of its beliefs that in fact should have prevented that conclusion, it is only at that later time that it can be withdrawn, and this is what active logic makes possible. In principle, in the limit, active logic should, in special cases at least, provide the same default conclusions as standard NMR formalisms; this is a topic of current investigation.

Several example problems were solved this way in real time. For instance, during the planning, new information could become available in contradiction with existing beliefs. In that work, contradictions were treated in conjunction with default rules, where the rule that "P follows by default from Q" can be represented as in 5:

$$\begin{array}{ll} \texttt{i} & : \\ \texttt{i+1} & : \end{array} \quad \frac{\texttt{Q, -Know(-P,i), Now(i)}}{\texttt{P}} \tag{5}$$

Thus if ¬P is not known at the current time, and if Q is known, then P is inferred by default at the next time step. However, it may turn out that at a later time, evidence for ¬P becomes known and a contradiction results. In the past work the particular example problems allowed for a very simple expedient in such cases: disinherit the default conclusion and accept the non-default evidence.

But while disinheriting contradictands is a reasonable first step, it is often not enough even to "defuse" the contradiction for long. P and ¬P may have come into KB for reasons that are still in force and the system may re-derive P and ¬P, or other similar conflicts, later on. Thus, in [Miller and Perlis, 1993; Gurney *et al.*, 1997; Purang, 2001] we have investigated ways to allow an active logic-based reasoner to retrace its history of inferences, examine what led to the contradiction, and perform metareasoning concerning which of these warrants continued belief.

However, in general such an expedient is far too naive to be useful, and instead more sophisticated conflict-resolution methods are needed. Current research is aimed at the development of a *typology* of contradictions, which will allow appropriately specific methods to be applied to individual cases [Anderson and Perlis, forthcoming]. This in turn will provide for much more useful real-time deadline planning in which new evidence can be weighed against old along multiple dimensions.

Finally, we extend the logic with one special proposition, *call*, which, if it is ever proved, will initiate external action (that can be reasoned about and tracked through observation). Our current implementation of active logic, as represented in ALMA [Purang *et al.*, 1999], already has this ability to initiate, observe and respond to external events and non-logical processes.

4 Examples

Active logic forms the reasoning core of our task-oriented dialog agent, ALFRED (Active Logic For Reason Enhanced Dialog). To reduce the complexity of the dialog problem, and in accordance with our model of buying a train ticket in a foreign country, ALFRED is always connected to given domain. What there *is* in that domain, and what ALFRED can *do* in that domain is defined for ALFRED in the form of a domain-oriented command language. ALFRED's ultimate task, then, is to translate a user's utterance into an appropriate statement in the command language of the target domain. If ALFRED encounters problems along the way, he must recognize them, and take appropriate steps to rectify the situation. Among ALFRED's current abilities are the following [Josyula *et al.*, 2003]:

1. Maintaining context

   ALFRED maintains the context of the ongoing conversation by keeping track of user intentions (interpretations of past utterances), needs and expectations. A "need" is created when there is some requirement (like determining the meaning of a word) that has to be met before the system can interpret a user utterance. An "expectation" is created when the system expects a particular kind of response from the user.

2. Introspection

   ALFRED can check whether it "knows" something by trying to prove that something. For instance, if ALFRED wants to check whether it knows the meaning of the word "Metro", it will try to prove meaning("Metro",X) where X is bound.

3. Identifying miscommunication

   ALFRED recognizes miscommunication problems by looking for contradictions in its interpretations of the user's intentions. In an example originally described in [Traum *et al.*, 2002], the user initially says:

   "Send the Boston train to New York"

   But the system moves a train other than the one the user intended by the phrase "the Boston train". Thus the user says:

   "No, send the *Boston* train to New York"

   This creates a contradiction—do X, don't do X—which the system must recognize and consider. In order for the system to properly interpret the correction in the latter portion of the above utterance, it must come to recognize that "no" is not a change of mind on the user's part (as it might have been), nor is it an incoherent self-contradiction by the user (don't send it and do send it), but rather an implicit correction of the intervening action taken by the system (sending, say, the Burlington train instead of the Boston train). Recognizing this, the system reconsiders its initial interpretation of the phrase "Boston train", and chooses another appropriate candidate to move.

4. Utterance generation

   ALFRED informs the user of the action that it is taking in response to a user utterance, be it an internal action or a domain action. This kind of communication helps in grounding [Brennan, 1998; Clark and Schaefer, 1987; 1989b; Clark and Brennan, 1991] so that the user knows whether the user's intention has been discerned correctly.

5. Using meta-dialog

   ALFRED is capable of engaging in meta-dialog with the human user when necessary, in order to identify communication problems and perform dialog repairs. Whereas in the above example the repair proceeded without additional user input, there are cases where the best way to a solution involves asking the user for help. Consider the following exchange:

   USER: Switch on the living room lamp.
   ALFRED: There are two lamps in the living room. Which one do you mean?
   USER: The one on the table.
   ALFRED: OK. (ALFRED turns on the light)

6. Reference resolution

   Reference resolution involves identifying the objects implied in a user utterance. In the simplest case, the reference would be the domain name of an object (like *Metroliner*). In more complicated cases, the reference resolution would involve disambiguation of the reference using introspection or meta-dialog with user.

7. Learning new words

   ALFRED can learn new names that refer to already existing objects in the domain. It can also learn new ways to accept known commands. This feature can allow a given user to negotiate a more comfortable vocabulary for interaction, and helps to increase the vocabulary of the system as a whole. This is ilustrated by the following exchange:

   USER: Shoot the Bullet to Boston.
   ALFRED: I do not know the command "shoot". What does "shoot" mean?
   USER: "Shoot" means "send".
   ALFRED: OK. (ALFRED sends the Bullet train to Boston)

8. Understanding the use-mention distinction

   ALFRED is sensitive to whether a word is being used or mentioned, and interprets the relevant utterance accordingly. In the exchange above, ALFRED recognizes that when the user defines "shoot", the word is being mentioned, and interprets the utterance accordingly. Following [Saka, 1998], we have chosen to characterize the use-mention distinction in terms of the possible ostensions of words. That is, we consider X is being "used", if the speaker intends to direct the thoughts of the audience to the extension of X; and X is being "mentioned", if he intends to direct the thoughts of the audience to some item associated with X *other than its extension.* See [Anderson *et al.*, 2002] for more details. The features that ALFRED uses to make this distinction are context, cues and meta-dialog.

9. Connecting to different domains

   As mentioned above, ALFRED functions as a kind of translator between human natural language and the specialized language of the task-oriented domain. This specialized language can vary from menu driven commands in the simplest case to natural language-like commands in a more complicated scenario. However, although the domain must be defined in this way for ALFRED, the mechanisms for translation, error detection, and correction remain the same from domain to domain. This makes orienting ALFRED to a different domain is as easy as changing the files specifying the specialized language for that domain.

   The upshot of this is that ALFRED can act as a natural language interface between a user and *any* task-oriented system, thereby enhancing the performance of the original system interface by incorporating a suite of dialog error detection and repair strategies.

5 Future Directions

Our primary claim, illustrated by our dialog system ALFRED, is that one can enhance the interactive capability of a task-oriented computer system by adding the ability to detect and recover from miscommunication problems, including ambiguous references, incompatible or contradictory user intentions, and the use of unknown words.

This technology can already be usefully applied to current application domains, such as home-control software, and we expect that the techniques employed can be refined and extended to handle more sophisticated domains.

We are working on methods of structuring and manipulating conceptual relations to allow ALFRED to learn not just new words for known objects or actions, but also genuinely new objects and concepts. We are also continuing to study human dialog behavior for clues as to what strategies can be employed by competent language users to ensure adequate communication even under difficult circumstances. Our long range goal is to be able to emulate the various behaviors and clarification devices employed by a human learning a foreign language or a novice learning a new subject.

References

[Andersen *et al.*, 1999] C. Andersen, D. Traum, K. Purang, D. Purushothaman, and D. Perlis. Mixed initiative dialogue and intelligence via active logic,. In *Proceedings of the AAAI'99 Workshop on Mixed-Initiative Intelligence*, 1999.

[Anderson and Perlis, forthcoming] Michael L. Anderson and Don Perlis. Detecting, classifying, and handling contradictions in a large, dynamic information environment, *forthcoming.* University of Maryland, College Park.

[Anderson *et al.*, 2002] Michael L. Anderson, Yoshi Okamoto, Darsana Josyula, and Don Perlis. The use-mention distinction and its importance to HCI. In *Proceedings of the Sixth Workshop on the Semantics and Pragmatics of Dialog*, 2002.

[Anderson *et al.*, forthcoming] Michael L. Anderson, Don Perlis, Andrew Fister, Bryant Lee, and Danny Wang. On the frequency and types of meta-language in conversation: a preliminary report, *forthcoming.* University of Maryland, College Park.

[Bohlin *et al.*, 1999] Peter Bohlin, Johan Bos, Staffan Larsson, Ian Lewin, Colin Matheson, and David Milward. Survey of existing interactive systems. *TRINDI Technical Report*, D1.3:1–57, 1999.

[Brennan and Hulteen, 1995] Susan E. Brennan and Eric A. Hulteen. Interaction and feedback in a spoken language system: A theoretical framework. *Knowledge-Based Systems*, 8:143–151, 1995.

[Brennan, 1998] Susan E. Brennan. The grounding problem in conversations with and through computers. In S.R. Fussell and R.J. Kreuz, editors, *Social* *and Cognitive Psychological Approaches to Interpersonal Communication*, pages 201–225. Lawrence Erlbaum, 1998.

[Brennan, 2000] Susan E. Brennan. Processes that shape conversation and their implications for computational linguistics. In *Proceedings of the 38th Annual Meeting of the Association for Computational Lingusitics*, 2000.

[Brown, 1987] F. Brown, editor. *The Frame Problem in Artificial Intelligence.* Morgan Kaufmann, 1987.

[Cahn and Brennan, 1999] Janet E. Cahn and Susan E. Brennan. A psychological model of grounding and repair in dialog. In *Proceedings of the AAAI Fall Symposium on Psychological Models of Communication in Collaborative Systems*, pages 25–33, 1999.

[Cassell *et al.*, 2000] Justine Cassell, Joseph Sullivan, Scott Prevost, and Elizabeth Chuchill. *Embodied Conversational Agents.* MIT Press, Cambridge, MA, 2000.

[Clark and Brennan, 1991] H.H. Clark and Susan E. Brennan. Grounding in communication. In J. Levine L.B. Resnik and S.D. Teasley, editors, *Perspectives on Socially Shared Cognition*, pages 127–149. 1991.

[Clark and Schaefer, 1987] H.H. Clark and E.F. Schaefer. Collaborating on contributions to conversations. *Language and Cognitive Processes*, 2:19–41, 1987.

[Clark and Schaefer, 1989a] Herbert H. Clark and Edward F. Schaefer. Contributing to discourse. *Cognitive Science*, 13:259–294, 1989. Also appears as Chapter 5 in [Clark, 1992].

[Clark and Schaefer, 1989b] H.H. Clark and E.F. Schaefer. Contributing to discourse. *Cognitive Science*, 13:259–294, 1989.

[Clark, 1992] Herbert H. Clark. *Arenas of Language Use.* University of Chicago Press, 1992.

[Elgot-Drapkin and Perlis, 1990] J. Elgot-Drapkin and D. Perlis. Reasoning situated in time I: Basic concepts. *Journal of Experimental and Theoretical Artificial Intelligence*, 2(1):75–98, 1990.

[Gärdenfors, 1988] P. Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States.* MIT Press, Cambridge, MA, 1988.

[Gurney *et al.*, 1997] J. Gurney, D. Perlis, and K. Purang. Interpreting presuppositions using active logic: From contexts to utterances. *Computational Intelligence*, 1997.

[Hobbs, 1985] Jerry Hobbs. Ontological promiscuity. In *Proceedings ACL-85*, pages 61–69, 1985.

[Hwang and Schubert, 1993] C. H. Hwang and L. K. Schubert. Episodic Logic: A comprehensive, natural representation for language understanding. *Minds and Machines*, 3:381–419, 1993.

[Josyula *et al.*, 2003] Darsana Josyula, Michael L. Anderson, and Don Perlis. Towards domain-independent, task-oriented, conversational adequacy, 2003. IJCAI-2003 Intelligent Systems Demonstrations.

[Krahmer *et al.*, 1999a] Emiel Krahmer, Marc Swerts, Mariet Theune, and Mieke Weegels. Error detection in spoken human-machine interaction. In *Proceedings of Eurospeech'99*, Budapest, Hungary, 1999.

[Krahmer *et al.*, 1999b] Emiel Krahmer, Marc Swerts, Mariet Theune, and Mieke Weegels. Problem spotting in human-machine interaction. In *Proceedings of Eurospeech'99*, Budapest, Hungary, 1999.

[McCarthy and Hayes, 1969] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, pages 463–502. Edinburgh University Press, 1969.

[McRoy *et al.*, 1997] Susan W. McRoy, Susan Haller, and Syed Ali. Uniform knowledge representation for language processing in the b2 system. *Journal of Natural Language Engineering*, 3(2/3):123–145, 1997.

[Miller and Perlis, 1993] M. Miller and D. Perlis. Presentations and this and that: logic in action. In *Proceedings of the 15th Annual Conference of the Cognitive Science Society*, Boulder, Colorado, 1993.

[Nirkhe *et al.*, 1997] M. Nirkhe, S. Kraus, M. Miller, and D. Perlis. How to (plan to) meet a deadline between *now* and *then. Journal of logic computation*, 7(1):109–156, 1997.

[Paek and Horvitz, 1999] Tim Paek and Eric Horvitz. Uncertainty, utility and misunderstanding: A decision-theoretic perspective on grounding in conversational systems. In *Proceedings, AAAI Fall Symposium on Psychological Models of Communication in Collaborative Systems*, 1999.

[Perlis *et al.*, 1998] D. Perlis, K. Purang, and C. Andersen. Conversational adequacy: mistakes are the essence. *Int. J. Human-Computer Studies*, 48:553–575, 1998.

[Poesio and Traum, 1997] Massimo Poesio and David R. Traum. Conversational actions and discourse situations. *Computational Intelligence*, 13(3), 1997.

[Purang *et al.*, 1999] K. Purang, D. Purushothaman, D. Traum, C. Andersen, D. Traum, and D. Perlis. Practical reasoning and plan execution with active logic. In *Proceedings of the IJCAI'99 Workshop on Practical Reasoning and Rationality*, 1999.

[Purang, 2001] K. Purang. *Systems that detect and repair their own mistakes.* PhD thesis, Department of Computer Science, University of Maryland, College Park, Maryland, 2001.

[Purver *et al.*, 2002] Matthew Purver, Jonathan Ginzburg, and Patrick Healey. On the means for clarification in dialogue. In R. Smith and J. van Kuppvelt, editors, *Current and New Directions in Discourse and Dialogue.* Kluwer Academic Publishers, 2002.

[Purver, 2002] Matthew Purver. A clarification request markup scheme for the BNC. Technical Report TR-02-02, Department of Computer Science, King's College London, February 2002.

[Saka, 1998] Paul Saka. Quotation and the use-mention distinction. *Mind*, 107:113–135, 1998.

[Traum *et al.*, 1996] David R. Traum, L. K. Schubert, M. Poesio, N. G. Martin, M. Light, C. H. Hwang, P. Heeman, G. Ferguson, and J. F. Allen. Knowledge representation in the TRAINS-93 conversation system. *International Journal of Expert Systems*, 9(1):173–223, 1996.

[Traum *et al.*, 2002] David Traum, Carl Andersen, Yuan Chong, Darsana Josyula, Michael O'Donovan-Anderson, Yoshi Okamoto, Khemdut Purang, and Don Perlis. Representations of dialogue state for domain and task independent meta-dialogue. *Electronic Transactions on Artificial Intelligence*, 6, 2002.

[Traum, 1994] David Traum. *A Computational Theory of Grounding in Natural Language Conversation.* PhD thesis, University of Rochester, 1994.