

A Practical, Declarative Theory of Dialog*

Susan W. McRoy (mcroy@uwm.edu) and Syed S. Ali (syali@uwm.edu)
University of Wisconsin–Milwaukee
Milwaukee, WI 53201

November 1, 1999

Abstract

The general goal of our work is to investigate computational models of dialog that can support effective interaction between people and computer systems. We are particularly interested in the use of dialog for training and education. To support effective communication, dialog systems must facilitate users' understanding by incrementally presenting only the most relevant information, by evaluating users' understanding, and by adapting the interaction to address communication problems as they arise.

Our theory provides a specification and representation of the linguistic, intentional, and social information that influence how people understand and respond in an ongoing dialog and an architecture for combining this information. We represent knowledge *uniformly* in a single, declarative, logical language where the interpretation and performance of communicative acts in dialog occurs as a result of reasoning.

1 Introduction

We are investigating computational models of dialog that can support robust, effective communication between people and computer systems [McRoy *et al.*, 1997, McRoy, 1995, McRoy, 1998, McRoy and Hirst, 1993, McRoy *et al.*, 1998a, McRoy *et al.*, 1998b, Ali *et al.*, 1999a, Restificar *et al.*, 1999a, Restificar *et al.*, 1999b]. Developing such methods requires:

- The specification and representation of information that affects learning and decision-making.
- The specification and implementation of algorithms for discriminating among alternatives, for recognizing errors, and for generating repairs.

The evaluation of this work involves the construction of computer programs that collaborate with people on tasks such as collaborative training or decision support. More specifically, we are developing collaborative tutoring systems for medical students to practice their decision-making skills (the B2 project) and for blood pressure health education (the ColTrain project).

The general model of processing for our work is one of an Intelligent Dialog System [Bordegoni *et al.*, 1997]. Intelligent Dialog Systems (IDS) are concerned with the effective management of an incremental, mixed-initiative interaction between the user and the system. This approach is in contrast with a presentation system, where the system's outputs are pre-planned (*e.g.* driven by a fixed plan or grammar) and not adapted to the user's apparent understanding or lack thereof. In an IDS, content to be presented, as well as the system's model of the user, change dynamically during an interaction.

Reasoning about dialog, such as to determine what a user's actions mean in the context of the dialog, whether a user's actions indicate understanding and agreement, and how to respond to a user's action,

*This work was supported by the National Science Foundation, under grants IRI-9701617 and IRI-9523666 and by a gift from Intel Corporation.

requires representing and combining many sources of knowledge. To support natural communication (which may contain fragments, anaphora, or follow-up questions), as well as to reason about the effectiveness of the interaction, a dialog system must represent both sides of the interaction; it must also combine linguistic, social, and intentional knowledge that underlies communicative actions [Grosz and Sidner, 1986, Lambert and Carberry, 1991, Moore and Paris, 1993, McRoy and Hirst, 1995]. To adapt to a user's interests and level of understanding (*e.g.* by modifying the questions that it asks or by customizing the responses that it provides), a dialog system must represent information about the user and the state of the ongoing task.

The architecture that we have been developing for building Intelligent Dialog Systems includes computational methods for the following:

- The representation of natural language expressions;
- The interpretation of context-dependent and ambiguous utterances;
- The recognition and repair misunderstandings (by either the system or the user);
- The detection and rebuttal of arguments; and
- The generation of natural language responses in real-time.

In what follows, we present an architecture and computational theory that addresses these issues. We present a detailed example of the representations and processing require to answer a question. We will then describe some of the components of our work and how they make use of the uniform, declarative representation of dialog.

2 A General Architecture for Dialog

Our architecture for Intelligent Dialog Systems is shown in Figure 1. The INPUT MANAGER and DISPLAY MANAGER deal with input and output, respectively. The input modalities will include typed text, spoken text, mouse clicks, and drawing. The output modalities will include text, graphics, speech and video. The DIALOG MANAGER is the component through which all input and output passes. This is important because the system must have a record of everything that occurred (both user and system-initiated). If the user chooses to input language, the LANGUAGE MANAGER is handed the text to parse and build the appropriate representation which is then interpreted by the dialog manager. The DOMAIN MANAGER component will be comprised of general rules of the task as well as specific information associated with how the CONTENT is to be presented. The content will be generated, prior to system use, by the use of AUTHORING TOOLS that allow the rapid development of the content. Based on the ongoing interaction, as well as information provided by the user, USER BACKGROUND & PREFERENCES are tracked. The status of the interaction is evaluated incrementally by the EVALUATION MANAGER, which affects the ongoing dialog and user model. We will present some of these components in more detail in Sections 4 and 5.

This architecture builds on our prior work, where the user is on a “thin” client personal computer interacting with a server that contains all the components described [McRoy *et al.*, 1997]. Most components of this architecture are general purpose; to retarget the system for a new domain, one would need to respecify only the domain and content.

All components within the large box on the right share a common representation language and a common inference and acting system.

2.1 Dialog Processing

Actions by the user are interpreted as communicative acts by considering what was observed and how it fits with the system's prior goals and expectations. First, a parser with a broad coverage grammar builds

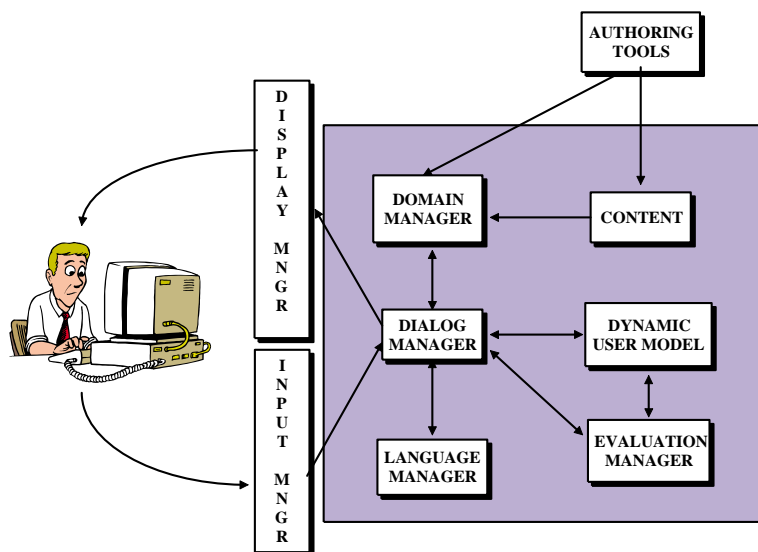


Figure 1: Our Architecture for Intelligent Dialog Systems

a mixed-depth representation of the user's actions.¹ This representation includes a syntactic analysis and a partial semantic analysis. Mixed-depth representations are constructed incrementally and opportunistically. They are used to address the ambiguity that occurs in utterances, without sacrificing generality. Encoding decisions that require reasoning about the domain or about the discourse context are left to subsequent processing.

Second, the dialog manager uses domain knowledge to map linguistic elements onto domain elements and to refine some semantic structures. This level of processing includes the interpretation of noun phrases, the resolution of anaphora, and the interpretation of sentences. For example, the mixed-depth representation leaves the possessive relationship uninterpreted; at this stage, domain information is used to identify the underlying conceptual relationship (*i.e.* ownership, part-whole, kinship, or object-property), as in the following:

The man's hat (ownership); the man's arm (part-whole); the man's son (kinship); the man's age (object-property).

Next, the dialog manager identifies higher-level dialog exchange structures and decides whether the new interpretation confirms its understanding of prior interaction. Exchange structures are pairs of utterances (not necessarily adjacent, because a subdialog may intervene) such as question-answer or inform-acknowledge. The interpretation of an exchange indicates how the exchange fits with previous ones, such as whether it manifests understanding, misunderstanding, agreement, or disagreement.

Finally, the assertion of an interpretation of an utterance triggers the appropriate actions (*e.g.* a question will normally trigger an action to compute the answer) to provide a response. In Section 3.2, we will illustrate our approach by working through the answer to a question: *What is Mary's age?*

¹A *mixed-depth representation* is one that may be shallow or deep in different places, depending on what was known or needed at the time the representation was created [Hirst and Ryan, 1992]. Shallow representations include a representation of the interaction such as a sequence of time-stamped events. Deep representations include conventional first-order (or higher-order) AI knowledge representation. (The distinction is similar to the one between locutionary and illocutionary acts [Austin, 1962].)

2.2 The Underlying Inference and Acting System

The inference and acting system provides services used for interpreting the user's actions and for constructing a response:

- It maintains a record of the events performed by the system and the user during their interaction (the discourse history).
- It keeps information needed by the system to select actions, including the relationships between plans and acts and between plans and goals.
- It represents and reasons over information about the domain and mediates between the analyzers and external knowledge sources.
- It partitions the knowledge for efficient reasoning.

All this information is represented as a propositional semantic network, that is, as a graph composed of nodes and labeled arcs, where the propositions are represented by the nodes. A propositional semantic network is a framework for representing the concepts of a cognitive agent who is capable of using language (hence the term *semantic*). The information is represented as a graph composed of nodes and labeled directed arcs. In a *propositional* semantic network, the propositions are represented by the nodes, rather than the arcs; arcs represent only non-conceptual binary relations between nodes.

The particular knowledge representation system that is used is SNePS [Shapiro and Rapaport, 1992]. SNePS provides facilities for building and finding nodes, as well as for (first- and second-order) reasoning, truth-maintenance, planning/acting, and knowledge partitioning (for user- and system-models). Our theory is knowledge-intensive, knowledge partitioning allows tractable inference in real-time.

```
* (describe (assert member "Tweety" class "bird"))
(M1! (CLASS bird) (MEMBER Tweety))
```

Figure 2: `Tweety is a bird` as represented in the knowledge base.

Case frames are used to represent propositions. Case frames are conventionally agreed upon sets of arcs emanating from a node. For example, to express the proposition that A *isa* B, we use the MEMBER-CLASS case frame, which is a node with a MEMBER arc and a CLASS arc. Figure 2 shows the construction and representation of `Tweety is a bird` as node M1!. An extensive collection of standard case frames is provided in [Shapiro *et al.*, 1994] and additional case frames can be defined as needed. We use many standard case frames as well as several new ones.

3 The Representation of Discourse

The dialog manager builds five levels of representation, shown in Figure 3, to capture the content, structure, and sequence of the system's and the user's utterances. Together these levels capture everything that the user and the system have said, as well as how their utterances extend the ongoing discourse. They are needed to allow the system to interpret context-dependent utterances such as *Why?* or to deal with unexpected utterances, such as misunderstandings or arguments. All this knowledge is represented uniformly in one knowledge base; however we partition the knowledge for efficiency.

The utterance level is a (mixed-depth) representation of what the user typed or selected with a mouse, as produced by the parser. The second level corresponds to the sequence of utterances, which enables the system to reason about temporal ordering constraints. (This level is comparable to the linguistic structure in the

| |
|-------------------------------------------|
| interpretation of exchanges |
| exchanges (pairs of interpretations) |
| system's interpretation of each utterance |
| sequence of utterances |
| utterance level |

Figure 3: Five Levels of Representation

tripartite model of [Grosz and Sidner, 1986]). The third level comprises the system's interpretation of each utterance. Each utterance event (from level 1) will have an associated system interpretation, which corresponds to a communicative act (such as question or command) which may reference entities from the underlying task. The fourth and fifth levels of the discourse model are exchanges and interpretations of exchanges, respectively. These levels represent a key difference between our work and previous approaches. These structures are determined on the basis of a number of domain-independent schemata which are represented declaratively in a logical language. The starting point for these schemata was sociolinguistic accounts of dialog [Schegloff, 1992, Clark and Schaefer, 1989] and Grice's [Grice, 1975] notion of reflexive intentions. Most AI approaches to dialog are based on Searle's [Searle, 1969, Searle, 1979] account of speech acts and STRIPS [Fikes and Nilsson, 1971, Sacerdoti, 1977] style plan operators of traditional AI. Our approach is more flexible and can better adapt to failed expectations (such as in misunderstanding or argumentation).

3.1 The Levels in an Example Dialog

We illustrate some (but not all) of the levels of our dialog model for the dialog below:

| | | |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------|---|
| User: | Why does a positive HIDA suggest gallstones? | 1 |
| B2 | <i>In the case of Mr Jones, the pretest probability of gallstones is 0.135. A positive HIDA test results in a post-test probability of 0.307.</i> | 2 |
| User: | I mean for what reason. | 3 |
| B2: | <i>Oh. HIDA detects cholecystitis, which is caused by gallstones.</i> | 4 |

Figure 4 illustrates the utterance sequence, interpretation, and exchange levels of representation that would result after this conversation.² Starting from the top of the figure, we see the following:

- Nodes M150, M160, and M170 represent the sequencing relations that hold between utterances 1–4. (The first row of boxes gloss the subnetworks corresponding to the utterance representations.) The sequence nodes would be constructed using the utterance level representations produced by the parser.
- Nodes M141, M151, M161, and M171 represent the interpretations that B2 gives to utterances 1–4, at the time that it parsed them. (The second row of boxes gloss the subnetworks corresponding to the interpretations.) The interpretations themselves correspond to representations of the speaker's actions on the domain or the discourse; these interpretations are derived from the utterance representations

²Currently, our system produces simplistic natural language output, rather than the output shown. We are still working on improving the text planning part of our system.

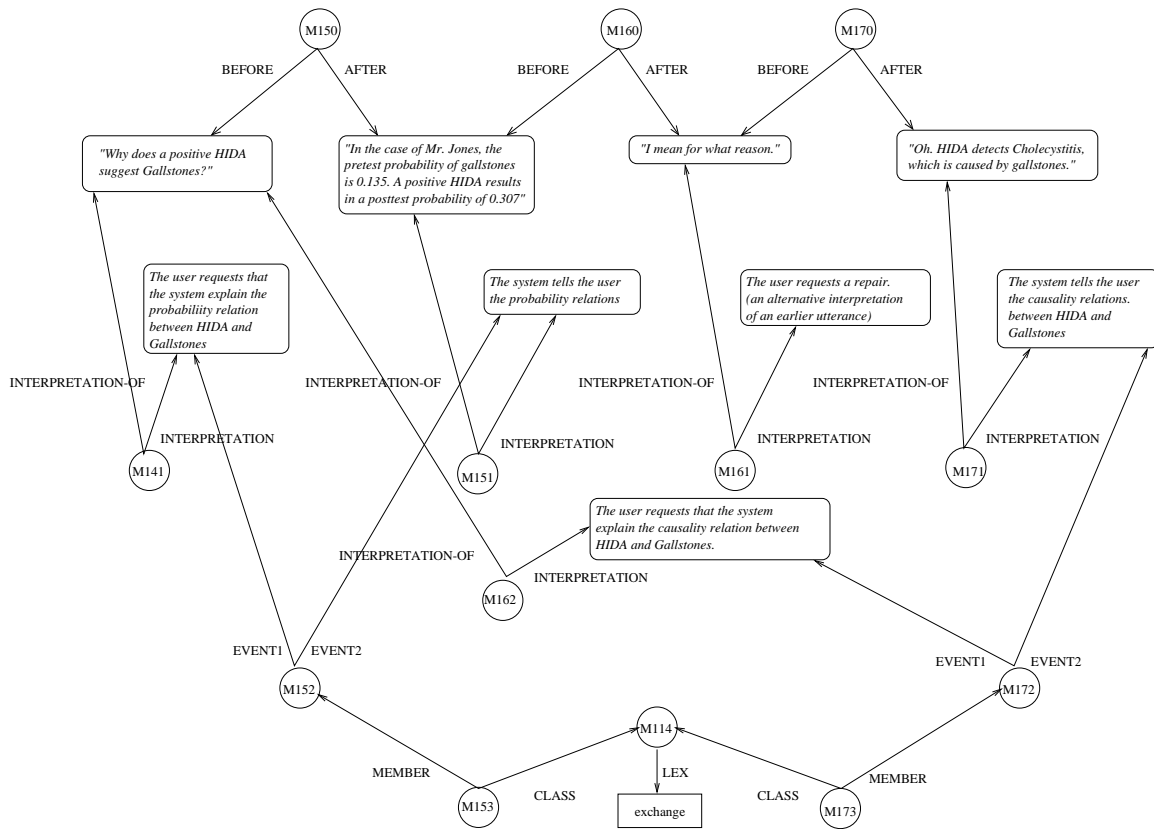


Figure 4: This figure shows (a gloss of) the utterance level representations, the utterance sequence-level representations, (a gloss of) the interpretation level representations, and the exchange-level representations for a dialog containing a repair.

on the basis of linguistic information, social conventions, and domain-specific plans. For example, to derive the interpretation of M141, the system reasons that when the user asks a why-question about an unspecified relation between two concepts in the Bayesian network, then it can be interpreted as a **request** for the system to perform the action **describe-probability-chain** on the two nodes. (Other interpretations are possible, the system need only find one that applies.) In the terminology of the previous section, M141 is explained as a case of plan-adoption, M151 is explained as acceptance, M161 is a self-misunderstanding (of utterance 1), M171 is a repair, which also acts as the acceptance for the interpretation of M141.

- Nodes M152 and M172 identify the exchange structure of utterances 1 and 2 (the original interpretation) and the exchange structure of utterances 1 and 4 (the repaired interpretation). When a node is recognized as the acceptance of another node, those two nodes can be taken as a complete exchange. In the figure, M153 and M173 are propositions that the two structures, respectively, are indeed exchanges.

The exchange structure is significant because, the system will step back through this structure if it needs to reason about alternative interpretations. The exchange structure indicates how each speaker displayed their understanding of the other’s previous utterances. (The utterance sequence will not always provide this information because exchanges can be nested inside each other, *e.g.* to ask a clarifying question.)

The interpretation of M161 is special, because it neither begins a new exchange nor completes an open one. (This would be determined by its linguistic form and by the expectations created by the previous interaction.) When the system fails to find a domain plan (*e.g.* a request to display part of the network) or a discourse plan (*e.g.* a request to clarify a previous utterance or an answer to a question from the system), then it considers evidence of failure. In this case, the surface form of the utterance suggests looking back in the conversation to consider an alternative domain plan. Finding an utterance that admits an alternative interpretation—corresponding to an alternative domain plan—a new interpretation (M162) is constructed, which results in a repair action by the system to accept it. (If there had been no alternative domain plan, then the system would not be able to form any interpretation of utterance 3—a case of non-understanding—and would subsequently ask the user to provide more information about the problem.)

The result of dialog processing is thus a detailed network of propositions that indicates the content of the utterances produced by the system or the user, their role in the interaction, and the system’s belief about what has been understood. If necessary, the system will be able to explain why it produced the utterances that it did and recover from situations where communication has failed.

3.2 A Detailed Example with Representations

Computationally, the system processes dialog by parsing communicative acts into mixed-depth representation, the construction of these representations triggers inference to determine an interpretation, and finally the derivation of an interpretation triggers an acting rule that performs an action that satisfies the user and system intentions.

To illustrate, we will now consider the underlying representations that are used when processing the question: *What is Mary’s age?*. The steps that occur in answering this question are:

- The parser produces a mixed-depth representation of the utterance (where the utterance is assigned the discourse entity label B4).
- The addition of the mixed-depth representations triggers inference to:
 1. Invoke content interpretation rules, to deduce that age is an attribute of *Mary*, and that the utterance is about an object-property relationship (between *what* and *Mary’s age*).
 2. Invoke anaphora interpretation rules to find a known entity named *Mary*.

3. Invoke pragmatic interpretation rules that derive that the communicative act associated with the utterance is an *askref* and that it initiates a new (question-answer) exchange.

- The resulting interpretation of the question triggers an acting rule that answers the question.
- Finally, this leads to a goal whose plan calls for the system to say *42* (the answer).

All interpretation and acting is done with the same representation language, thus a complete record of all of these events is maintained. We now consider this example in more detail, showing most (but not all, for space reasons) of the representation(s) used.

3.2.1 Parsing, content, and anaphora interpretation

As mentioned above, the question is parsed by a broad-coverage grammar which builds the utterance-level (mixed-depth) representation(s) as shown in Figure 5. For clarity, the semantic networks are shown as simplified feature structures. Propositions are labeled as *Mj* and (potential) discourse entities are labeled as *Bk*. In Figure 5, three propositions are produced from the initial parse of the question. Proposition M10 represents the fact that there was an utterance whose label is B4, whose form and attitude was interrogative copula, and whose content (M9) is some unknown *is* relation between B2 and B1. B1 corresponds to the pronoun *what* and B2 to *age*. Proposition M4 states that B2 is a member of the class of *age*. Finally, proposition M5 represents the fact that there is an unknown possessive relationship between B2 (an *age*) and B3 (an entity whose proper name is *Mary*).

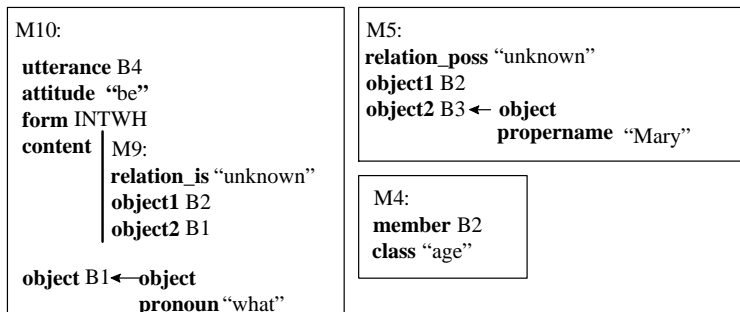


Figure 5: Initial Mixed-Depth Representation of: *What is Mary's age?*

As can be seen from Figure 5, the utterance-level propositions produced by the parser are the weakest possible interpretations of the utterance. Any question of this form would parse into similar utterance-level propositions; the subsequent interpretation(s) would vary.

In the next step of interpretation, M5 is further interpreted as specifying an attribute (B2, *i.e.* *age*) of an object (B3, *i.e.* *Mary*). This is a domain-specific interpretation and is deduced by an interpretation rule (not shown here for space reasons). The rule encodes that *age* is an attribute of an entity (and is not, for example, an ownership relation as in *Mary's dog*).

Figure 6 shows the interpretation rule used to deduce a partial interpretation of the utterance B4. A partial interpretation of an utterance is a semantic interpretation of the content of the utterance, apart from its communicative (pragmatic) force. This relationship will also be represented explicitly as a deep-surface relationship, which is derived using the rule shown in Figure 7.³ In addition, a separate rule (not shown) will

³Elements of the deep-surface relation may also be asserted as part of the domain knowledge, to express differences in terminology among utterances of the user, *e.g.* high blood pressure, and concepts in the domain, *e.g.* hypertension.

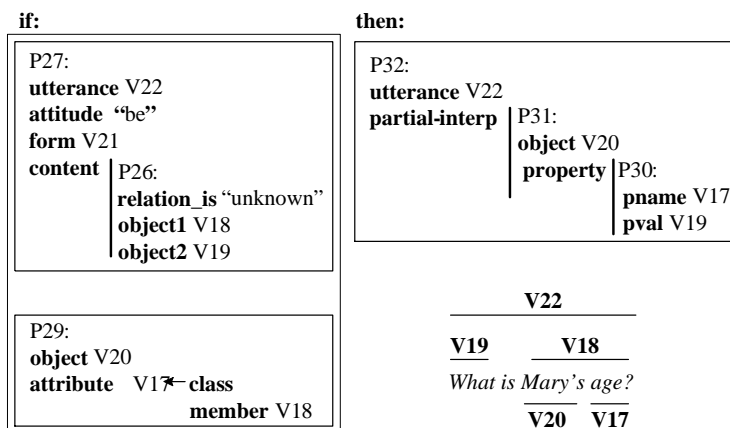


Figure 6: Partial Interpretation Rule for the Utterance B4

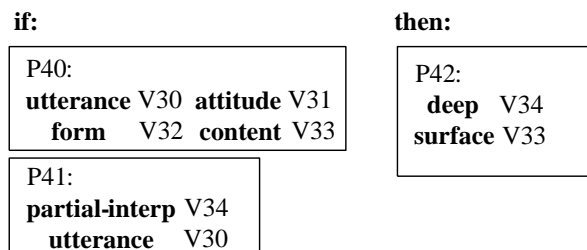


Figure 7: Derivation Rule for Making Explicit the Relation between an Utterance's Content and Partial Interpretation

be used to establish an equivalence relationship between B3 (the Mary mentioned in the utterance) and B0 (the Mary known to the system).⁴ As a result of the rule in Figure 6, the semantic content of the utterance is interpreted as an object-property relationship (pragmatic processing, discussed in the next subsection, will determine that the force is as a particular subclass of question *askref*).

In a rule such as in Figures 6 and 7, variables are labeled as Vn and, for clarity, the bindings of the variables of the rules are shown relative to the original question in the lower right corner. The *if* part of the rule in Figure 6, has two antecedents: (1) P27, requires that there be an copula utterance whose content is an unknown *is* relation between an entity (V19 *i.e.* *What*) and another entity (V18), (2) P29, requires that the latter entity (V18 *i.e.* *age*) is an attribute of another entity (V20 *i.e.* *Mary*). The consequent of this rule P32 stipulates that, should the two antecedents hold, then a partial interpretation of the utterance is that V20 (*i.e.* *Mary*) has a property whose name is V17 (*i.e.* *age*) and whose value is V19 (*i.e.* *what*). The rule of Figure 6 allows the interpretation of the mixed-depth representations of Figure 5 as a proposition, which expressed in a logical formula, is *has-property*(*Mary*, *age*, *what*)

⁴Currently, the system makes the simplifying assumption that all objects with the same name are the same entity; we are exploring rules for anaphora.

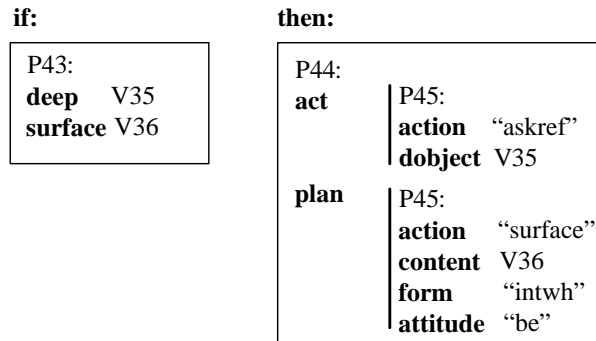


Figure 8: Text Planning Rule

3.2.2 Pragmatic interpretation

After parsing and content interpretation, discussed above, the system will have constructed Levels 1 and 2 of the discourse model and will be ready to consider the construction of the third level. At the third level, there might be several possible interpretations, but only one will be believed (which one is believed will be determined by inferences performed during the construction of the fourth and fifth levels of the discourse model). A communicative action is a possible interpretation of the user's literal action if the system believes that user's action is one of the known ways of performing the communicative act. We consider two rules, shown in Figures 8 and 9 that the system uses to derive a possible interpretation.

The rule in Figure 8 specifies the relationship between an utterance and the way it may be realized as an utterance. In this case, whenever there is a deep-surface relationship between two propositions V35 and V36, that is, V36 is a representation of how the user might express a proposition and V35 is a representation of how the system represents the concept in its model of the domain, then an agent (either the system or the user) may perform an *askref*⁵ by performing the (linguistic) action called "surface" to output the content V36 with a surface syntax of "intwh" and attitude "be". We call this type of rule a "text planning rule" because it may be used by the system either to interpret an utterance by the user or to generate an utterance by the system.

Figure 9 is a rule that specifies a possible interpretation of an utterance. It says that if a speaker makes an utterance, and that utterance is part of a plan that accomplishes an action, then an interpretation of the utterance is that the speaker is performing the action. This rule relies on the results of the text planning rule mentioned above, where P52 is matched against a text plan whose act is the following:

```
(M23 (ACTION "askref")
      (DOBJECT (M24 (OBJECT B0)
                    (PROPERTY (M25 (PNAME "AGE")
                                   (PVAL B1)))))))
```

and P50 is matched against the output of the parser with `form = intwh`, `attitude = be`, and

```
content = (M9 (RELATION_IS "unknown")
             (OBJECT1 B2)
             (OBJECT2 B1))
```

⁵An *askref* is a type of communicative act that is used to ask for the referent of some expression, akin to asking for the hearer's binding of some variable.

3.2.3 Answering the question

The assertion of an interpretation of the utterance as an **askref** and its acceptance as a coherent continuation of the dialog leads to an action by the system to answer the question.

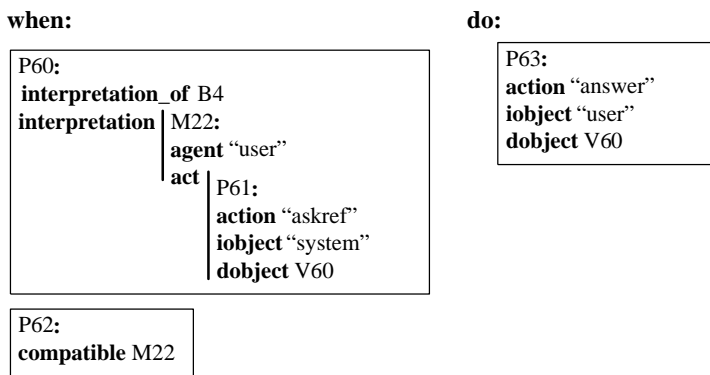


Figure 11: An Acting Rule for Responding to a Question

Figure 11 is an acting rule (by contrast to the inference rules discussed previously), which glosses as: if the user asks the system a question (P60) and the system believes that it is compatible with the dialog to answer the question (P62) then do the action of answering the question.⁶ To achieve the latter action (**answer**) the system uses a plan in which the system deduces possible values of *Mary's age* by replacing the **what** in the question with a variable, and responds by saying the answer (if any). This answer is realized as a natural language expression using our real-time generator, YAG (discussed in Section 5).

4 Representations Used in the Interpretation of Exchanges

We have been working on computational methods for recognizing and repairing misunderstandings (RRM) and for detecting and rebutting arguments (ARGUER). The methods are represented declaratively by means of rule schemata that are used by the underlying inference system to make meta-inferences about the interpretations that it has performed.

4.1 RRM

RRM (The Recognition and Repair of Speech Act Misunderstandings) [McRoy and Hirst, 1995, McRoy, 1998] provides a unified account of speech-act production, interpretation, and repair. These tasks are essential to the management of intelligent dialogs, because opportunities for errors in communication are unavoidable:

- The user's attention might not be focused on the aspect of the presentation that the system expects.
- The user might not have the same understanding of what a verbal description or a graphical image is meant to convey.

⁶Compatibility is a notion that is related to the coherence of dialog and the expression of reflexive intentions (see [McRoy and Hirst, 1995]). In this case, a question expresses the lack of knowledge about some referent and an intention to know it. This interpretation of the original utterance is compatible because the neither the user nor the system has indicated that the user already knows the answer—which might be the case, if, for example, the system had previously answered the question. If one interpretation is incompatible, another response might be possible (*e.g.* the generation of a repair), but that possibility is beyond the scope of this paper.

- The user might lack some of the requisite knowledge of the domain necessary to interpret a proposed explanation.

Thus, any computer system that communicates must be able to cope with the possibility of miscommunication. RRM addresses possible misunderstandings (as well as expected interpretations), while respecting the time-constraints of Intelligent Dialog Systems, by combining intentional and social accounts of interaction to capture the expectations that help constrain interpretation. An action is considered a manifestation of misunderstanding if there is no coherent link apparent and there is a reason for supposing that misunderstanding has occurred.

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>U1: Why does a positive HIDA suggest gallstones?</p> <p>S1: In the case of Mr. Jones, the pretest probability of gallstones is 0.135. A positive HIDA results in a posttest probability of 0.307.</p> <p>U2: I mean for what reason.</p> <p>S2: Oh. HIDA detects cholecystitis which is caused by gallstones .</p> | <p>Self-misunderstanding:</p> <p>Let s_1, s_2 be speakers; and u_1, u_2 be utterances by s_1 where u_1 precedes u_2.</p> <p>If interp_of(s_2, u_1, a_o) and interp_of(s_2, u_2, a_N) and a_N is not compatible with a_o, and textplan(u_1, a_o) and textplan(u_1, a_i) Then disbelieve (interp_of(s_2, u_1, a_o)) interp_of(s_2, u_1, a_i) mistake(s_2, a_o, a_i)</p> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

$s_1 = \mathbf{user}, \quad s_2 = \mathbf{system}, \quad u_1 = \mathbf{U1}, \quad u_2 = \mathbf{U2}$
 $a_o =$ User requested probability relations: **interp_of**(**system**, **U1**, a_o).
 $a_N =$ User requested a repair: **interp_of**(**system**, **U2**, a_N).
 $a_i =$ User requested causality relations: **interp_of**(**system**, **U1**, a_i).

Figure 12: A Schema for Detecting Misunderstanding

Figure 12 shows a schema for detecting a misunderstanding (in particular a misunderstanding that is detected by the agent who has misunderstood, *i.e.* a *self-misunderstanding* [Schegloff, 1992]). Figure 13 shows a schema for repairing a misunderstanding (in particular, for making a repair of a self-misunderstanding after hearing an unexpected reply, *i.e.* a *fourth-turn repair* [Schegloff, 1992]). (For clarity of presentation, we are not showing the detailed representations corresponding to these schema; they are similar to those of section 3.2.) ColTrain uses these RRM schemata to handle misunderstandings.

Figure 12 shows the dialog discussed in Section 3.1, the schema that is used to detect a misunderstanding and the bindings that are used when matching the schema against the knowledge base. The system's interpretations of the user's utterances U1 and U2 are not compatible. Additionally, there is an alternative interpretation of U1. This allows the system to decide that its original interpretation of U1 was incorrect. In summary, this schema allows the detection of the system's misunderstanding.

Figure 13 shows the same dialog, the schema that is used to repair a misunderstanding, and the bindings that are used when matching the schema against the knowledge base. This schema allows the system to perform a repair. In this case, the system has detected that it was a mistake to provide the probability relations and that the user wanted the causality relations. The repair is the (conventionally expected) reply to a request for causality information.

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>U1: Why does a positive HIDA suggest gallstones?</p> <p>S1: In the case of Mr. Jones, the pretest probability of gallstones is 0.135. A positive HIDA results in a posttest probability of 0.307.</p> <p>U2: I mean for what reason.</p> <p>S2: Oh. HIDA detects cholecystitis which is caused by gallstones .</p> | <p>Fourth-turn repair:</p> <p>Let s_p, s_2 be speakers;</p> <p>If mistake(s_2, a_o, a_p), and expect(s_p, s_2, a_p, a_R), and a_R is compatible with the dialog, Then Do($s_2, \text{repair}(a_R)$)</p> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

$s_1 = \text{user}$,
 $s_2 = \text{system}$,
 $a_o = \text{User requested probability relations: } \text{interp_of}(\text{system}, \text{U1}, a_o)$.
 $a_p = \text{User requested causality relations: } \text{interp_of}(\text{system}, \text{U1}, a_p)$.
 $a_R = \text{System informs user of causality relations: } \text{interp_of}(\text{system}, \text{S2}, a_R)$.

Figure 13: A Schema for Repairing Misunderstanding

4.2 ARGUER

Intelligent dialog systems must also be prepared to deal with argument. ARGUER (Argument Detection and Rebuttal) is an argumentation system that allows an Intelligent Dialog System to recognize when another agent disagrees with it and to attempt to convince the other agent to adopt the system’s view [Ali *et al.*, 1999a, Restificar *et al.*, 1999a, Restificar *et al.*, 1999b].

The method that we describe here, which is used in our system ARGUER, uses argument schemata that match the deep meaning representation of propositions that have been advanced in a dialog. In contrast to [Birbaum *et al.*, 1980, Vreeswijk, 1995, Zukerman *et al.*, 1998, Karacapilidis and Papadias, 1998, Alvarado, 1990], we use a general computational method of establishing relations between propositions. Argument schemata characterize important patterns of argument that are used to establish whether propositions *support* or *attack* other propositions. These patterns are instantiated by propositions expressed by the agents during a dialog, as well as related beliefs that the agents might hold. To account for disagreements, separate models of the agents’ beliefs are maintained, both for the system and the user. Hence, a proposition believed by the system might not necessarily be believed by the user. To generate a correct and convincing response, the system considers both its own beliefs and those beliefs held by the user. In addition to allowing for incremental processing of arguments, this method is *symmetric* because it can be used for interpretation or generation of arguments. This is important because the system can have the role of observer or participant.

When the user inputs an utterance, the system will attempt to interpret the user’s utterance as an attack or support on a prior utterance of the system. It does so by asking, What does the user’s utterance attack? and second, What does the user’s utterance support? All reasoning to answer these questions occurs in the user’s belief model and makes use of all relevant knowledge sources therein [Ali *et al.*, 1999b]. When there is an argument, the system’s response will attempt to attack some previous utterances of the user (or, failing that, provide supporting arguments to prior system utterances).

The underlying principle for detecting arguments in ARGUER is to find a general case of an argument schema into which the meaning representation of an utterance can be matched. (Argument schema can also

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>S1: Uncontrolled high blood pressure can lead to heart attack.</p> <p>U1: But I feel healthy.</p> <p>S2: Unfortunately, there are no signs or symptoms that tell whether your blood pressure is elevated.</p> <p>U2: Ok, I'll get it checked.</p> | <p>Attacking a consequence: Let s_1, s_2 be speakers; and u_1, u_2 be utterances by s_1 and s_2, respectively, and u_1 precedes u_2. If $\text{interp_of}(s_1, u_1, \text{implies}(a_1, a_2))$, and $\text{interp_of}(s_1, u_2, a_3)$, and $\text{implies}(a_1, \text{not}(a_3))$ Then $\text{attacks}(s_1, a_3, \text{implies}(a_1, a_2))$</p> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

$s_1 = \text{system}, \quad s_2 = \text{user}, \quad u_1 = \text{S1}, \quad u_2 = \text{U1}$
 $a_1 = \text{User might have uncontrolled high blood pressure,}$
 $a_2 = \text{User might have a heart attack}$
 $\text{implies}(a_1, a_2) = \text{Uncontrolled HBP can lead to heart attack (S1)}$
 $a_3 = \text{User is healthy (U1)}$

Figure 14: A Schema for Detecting an Argument

be used to generate a rebuttal.)

The example shown in Figure 14 illustrates an argument and a schema that could be used to detect it. This schema detects that U1 is a potential attack to S1. If the system's interpretation of S1 implies a consequence that is not consistent with the system's interpretation of U1 then U1 is an attack on S1. In short, since uncontrolled high blood pressure can lead to a heart attack (which is not healthy) the user's utterance is an attack.

The use of argument schemata for argument detection and rebuttal allows argument relations between propositions to be established dynamically. Moreover, the method is incremental in that it allows processing of each piece of the utterance and uses only a part of the argument to continue.

5 Generating Natural Language Responses in Real-Time

YAG (Yet Another Generator) is our system for generating natural language in real-time, as required for an Intelligent Dialog System. YAG combines a template-based approach for the representation of text with knowledge-based methods for representing content (*i.e.*, inputs can be concepts or mixed-depth propositions along with optional annotations to specify syntactic constraints).

Templates are declarative representations of text structure. Each form in a YAG template is a rule that expresses how a surface constituent should be realized, given features present in the input. YAG can accept feature structures directly, or can map propositions represented as SNePS case frames onto their feature structure equivalents and select an appropriate template for their realization. YAG's approach to realization is practical, because it's speed does not depend on the number of template types that have been defined.

Inputs to YAG may include multiple propositions as well as a list of control features, as shown in Figure 5. When processing this input, YAG treats the first proposition as the primary proposition to be realized. YAG will map the MEMBER-CLASS proposition to the template shown in Figure 16. The control features, **form** = **decl** and **attitude** = **be**, are also used in selecting the template. (If the form had been **interrogative**, a template for generating a yes-no question would have been used.)

Prior to realization, a mapping from each type of proposition to the name of the corresponding template in a mapping table is specified. (This is the primary task in constructing a new knowledge representation

Pluto is a dog.

```

((M2 (CLASS "dog")
      (MEMBER B2))
 (M5 (OBJECT B2)
      (PROPERNAME "Pluto")))
((form decl)
 (attitude be) )
)

```

Figure 15: Example input to YAG

```

(EVAL member)
(TEMPLATE verb-form
  ((process "be")
   (person (member person))
   (number (member number))
   (gender (member gender))) )
(EVAL class)
(PUNC "." left) )

```

Figure 16: A member-class Template.

```

(member-class
 ((decl
  (be (template member-class)
       (slot-map ((class class)
                  (member member)) )
               (feature nil)
              )
 ))
)

```

Figure 17: A Simplified Mapping Entry for the member-class Case Frame.

realization component for other knowledge representations.) Each mapping entry provides a declarative specification for constructing a feature structure from the propositions and control features. A sample entry of a mapping table is given in Figure 17.

6 Summary

This research supports robust, flexible, multi-modal, mixed-initiative interaction between people and computer systems by combining techniques from language processing, knowledge representation, and human-machine communication.

This work is important because it specifies an end-to-end, declarative, computational theory that uses a uniform framework to represent the variety of knowledge that is brought to bear in collaborative interactions. Specifically:

- The mixed-depth representations that we use allow the opportunistic interpretation of vaguely articulated or fragmentary utterances.
- The five levels of the discourse model capture the content, structure, and sequence of dialog, along with their interpretations.
- The interpretation and generation of utterances involves the integration of linguistic, intentional, and social information.

References

- [Ali *et al.*, 1999a] Syed S. Ali, Susan W. McRoy, and Angelo C. Restificar. A Computational Theory of Argument in Dialog. 1999. In preparation.
- [Ali *et al.*, 1999b] Syed S. Ali, Susan W. McRoy, and Angelo C. Restificar. Relevance in Argumentation. 1999. In submission.
- [Alvarado, 1990] S. Alvarado. *Understanding Editorial Text: A Computer Model of Argument Comprehension*. Kluwer Academic, 1990.
- [Austin, 1962] John L. Austin. *How to Do Things with Words*. Oxford University Press, London, England, 1962. Reprinted 1975.
- [Birnbaum *et al.*, 1980] L. Birnbaum, M. Flowers, and R. McGuire. Towards an AI Model of Argumentation. In *Proceedings of the AAAI-80*, pages 313–315, Stanford, CA, 1980.
- [Bordegoni *et al.*, 1997] M. Bordegoni, G. Faconti, M. T. Maybury, T. Rist, S. Ruggieri, P. Trahanias, and M. Wilson. A standard reference model for intelligent multimedia presentation systems. In *Proceedings of the IJCAI '97 Workshop on Intelligent Multimodal Systems*, 1997.
- [Clark and Schaefer, 1989] Herbert H. Clark and Edward F. Schaefer. Contributing to discourse. *Cognitive Science*, 13:259–294, 1989.
- [Fikes and Nilsson, 1971] R. E. Fikes and Nils J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.
- [Grice, 1975] H. P. Grice. Logic and conversation. In P. Cole and J. L. Morgan, editors, *Syntax and Semantics 3: Speech Acts*. Academic Press, New York, 1975.
- [Grosz and Sidner, 1986] B. J. Grosz and C. L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12, 1986.
- [Hirst and Ryan, 1992] Graeme Hirst and Mark Ryan. Mixed-depth representations for natural language text. In Paul Jacobs, editor, *Text-Based Intelligent Systems*. Lawrence Erlbaum Associates, 1992.
- [Karacapilidis and Papadias, 1998] N. Karacapilidis and D. Papadias. Hermes: Supporting Argumentative Discourse in Multi-Agent Decision Making. In *Proceedings of the AAAI-98*, pages 827–832, Madison, WI 1998.
- [Lambert and Carberry, 1991] Lynn Lambert and Sandra Carberry. A tri-partite plan-based model of dialogue. In *29th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 47–54, Berkeley, CA, 1991.
- [McRoy and Hirst, 1993] Susan W. McRoy and Graeme Hirst. Abductive explanation of dialogue misunderstandings. In *6th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*, pages 277–286, Utrecht, The Netherlands, 1993.
- [McRoy and Hirst, 1995] Susan W. McRoy and Graeme Hirst. The repair of speech act misunderstandings by abductive inference. *Computational Linguistics*, 21(4):435–478, December 1995.
- [McRoy *et al.*, 1997] Susan McRoy, Susan Haller, and Syed Ali. Uniform knowledge representation for nlp in the b2 system. *Natural Language Engineering*, 3(2):123–145, 1997.

- [McRoy *et al.*, 1998a] Susan W. McRoy, Susan M. Haller, and Syed S. Ali. Mixed Depth Representations for Dialog Processing. In *Proceedings of Cognitive Science '98*, pages 687–692. Lawrence Erlbaum Associates, 1998.
- [McRoy *et al.*, 1998b] Susan W. McRoy, Alfredo Liu-Perez, and Syed S. Ali. Interactive Computerized Health Care Education. *Journal of the American Medical Informatics Association*, 5(4):76–104, 1998.
- [McRoy, 1995] Susan W. McRoy. Misunderstanding and the negotiation of meaning. *Knowledge-based Systems*, 8(2–3):126–134, 1995.
- [McRoy, 1998] Susan McRoy. Achieving robust human-computer communication. *International Journal of Human-Computer Studies*, 48:681–704, 1998.
- [Moore and Paris, 1993] Johanna Moore and Cécile Paris. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651–695, 1993.
- [Restificar *et al.*, 1999a] Angelo C. Restificar, Syed S. Ali, and Susan W. McRoy. ARGUER: Using Argument Schemas for Argument Detection and Rebuttal in Dialogs. In *Proceedings of User Modeling 1999*. Kluwer, 1999. To appear.
- [Restificar *et al.*, 1999b] Angelo C. Restificar, Syed S. Ali, and Susan W. McRoy. Argument Detection and Rebuttal in Dialog. In *Proceedings of Cognitive Science 1999*, 1999. To appear.
- [Sacerdoti, 1977] Earl D. Sacerdoti. *A Structure for Plans and Behavior*. American Elsevier, New York, 1977.
- [Schegloff, 1992] Emanuel A. Schegloff. Repair after next turn: The last structurally provided defense of intersubjectivity in conversation. *American Journal of Sociology*, 97(5):1295–1345, 1992.
- [Searle, 1969] John Searle. *Speech Acts*. Cambridge University Press, Cambridge, England, 1969.
- [Searle, 1979] John Searle. A taxonomy of illocutionary acts. In his *Expression and Meaning: Studies in the Theory of Speech Acts*, pages 1–29. Cambridge University Press, London, 1979. Previously published as “A Classification of Illocutionary Acts”, in *Language and Society*, 1975.
- [Shapiro and Rapaport, 1992] Stuart C. Shapiro and William J. Rapaport. The SNePS family. *Computers & Mathematics with Applications*, 23(2–5), 1992.
- [Shapiro *et al.*, 1994] Stuart Shapiro, William Rapaport, Sung-Hye Cho, Joongmin Choi, Elissa Feit, Susan Haller, Jason Kankiewicz, and Deepak Kumar. A dictionary of SNePS case frames, 1994.
- [Vreeswijk, 1995] G. Vreeswijk. IACAS: An Implementation of Chisholm’s Principles of Knowledge. In *Proceedings of the 2nd Dutch/German Workshop on Nonmonotonic Reasoning*, pages 225–234, 1995.
- [Zukerman *et al.*, 1998] I. Zukerman, R. McConachy, and K. Korb. Bayesian Reasoning in an Abductive Mechanism for Argument Generation and Analysis. In *Proceedings of the AAAI-98*, pages 833–838, Madison, Wisconsin, July 1998.