A Declarative Model of Dialog *

Susan W. McRoy **Syed S. Ali**
mcroy@uwm.edu, syali@uwm.edu
Natural Language and Knowledge Representation Research Group
Electrical Engineering and Computer Science
University of Wisconsin-Milwaukee

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 model 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.

Introduction

We are investigating computational models of dialog that can support robust, effective communication between people and computer systems. In particular, we are concerned with communication that may involve temporary differences in understanding or agreement. The evaluation of this work involves the construction of cognitive agents (computer programs) that collaborate with people on tasks such as collaborative training or decision support.

Our interest in intelligent tutoring systems began with Banter (Haddawy, Jacobson, & Jr. 1997), a simple tutoring shell that generates word problems and short-answer questions, without maintaining any model of the dialog. The system generates word problems using "canned" templates, with values taken from a database of cases that specify a patient's medical history, findings from physical examinations, and results of diagnostic tests. When the user requests a story problem, the system presents a case and then asks the user to select the most effective diagnostic test to either rule in or rule out the target disease. (The correct answer to such a problem is based on a statistical model of people who see their doctor complaining of medical pains. This model comprises joint probability relations among known aspects of a patient's medical history, findings from physical examinations of the patient, results of previous diagnostic tests, and the different candidate diseases.) After providing an answer to the problem, a student can then request an explanation of the correct answer. The system will respond by providing a trace of the probability calculations. Students can also add new cases to the data base or ask the system to select the best diagnostic test.

A preliminary (and informal) user study of the Banter system with students at the Medical College of Wisconsin revealed two important facts: First, students like the idea of being able to set up hypothetical cases and witness how different actions might (or might not!) affect the statistical likelihood of a candidate diagnosis. Second, students do not like, and will not use, a system that overwhelms them with irrelevant information, such as complete sentences when a single word would do or a long trace of a probability calculations when a simple statement about the causal relation between two events was desired.

To improve the usability of this system and its usefulness as an educational tool we aimed to provide more focussed answers to requests for an explanation as well as to provide a natural language dialog interface to the database of medical cases. In particular, we added a dialog model so that users could ask brief context-dependent questions, such as *"Why CT?"* or *"What about ultrasound?"* and receive brief, context-dependent responses.

In more recent work, we have been trying to adapt our basic tutoring architecture to a new domain. In particular we have been looking at the Blood Pressure Measurement Education Program curriculum of the Milwaukee Chapter of the

*This work has been supported by the National Science Foundation, under grants IRI-9701617 and IRI-9523666, and by Intel Corporation.

American Heart Association. This curriculum includes basic facts about high blood pressure, equipment for measuring blood pressure, and procedures for measuring blood pressure and for counselling clients. For example, we are using our dialog model (extended to detect and rebut arguments) to help teach students how to counsel clients who have elevated readings (Restificar, Ali, & McRoy 1999a; 1999b). Figure 1 includes a dialog produced by our system (S) when it receives inputs from the user (U). This type of example also involves our template-based natural language generation component, YAG (McRoy, Channarukul, & Ali 2000). We are also working on teaching dialogs that combine small blocks of text that are presented incrementally so that students can control the pace of the presentation and ask questions if necessary.

S `Have your blood pressure checked.`
U `There is no need.`
S `Uncontrolled high blood pressure can lead to heart attack, heart failure, stroke or kidney failure.`
U `But I feel healthy.`
S `Unfortunately, there are no signs or symptoms that tell whether your blood pressure is elevated.`
U `Ok, I'll check my blood pressure.`

Figure 1: Example Dialog from Blood Pressure Measurement Program Domain

Intelligent Dialog Systems

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 and how to respond to a user's action, 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 & Sidner 1986; Lambert & Carberry 1991; Moore & Paris 1993; McRoy & 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 include computational methods for the following:

- The representation of natural language expressions and communicative actions;
- 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 model that addresses these issues, focusing on the knowledge and reasoning that underly the first three tasks mentioned above.

Uniform, Declarative Representations of Knowledge

In our model, knowledge about expressions and actions and about understanding and agreement are represented uniformly, in a single, declarative, logical language where the interpretation and performance of communicative acts in dialog occurs as a result of reasoning. We term a knowledge representation *uniform* when it allows the representation of different kinds of knowledge in the same knowledge base using the same inference processes. In our work, we have a single knowledge representation and reasoning component that acts as a blackboard for intertask communication and cooperation (Shapiro & Rapaport 1992). We structure the knowledge by the "links" between facts in the knowledge base. Thus, although a concept might be realized as a graphic, a textual word, or a spoken word, all realizations would share a common underlying concept.

For example, our tutoring shell, B2, is comprised of three distinct, but interrelated tasks that rely on a variety of information sources. The tasks are:

- Managing the interaction between the user and B2, including the interpretation of context-dependent utterances.
- Reasoning about the domain, such as the relation between components of a medical case history and diseases that might occur.

- Meta-reasoning about the statistical reasoner and its conclusions, including an ability to explain the conclusions by identifying the factors that were most significant.

The tasks interact by addressing and handling queries to each other. However, the knowledge underlying these queries and the knowledge needed to generate a response can come from a variety of knowledge sources. Translating between knowledge sources is not an effective solution.

The information sources that B2 uses include:

- Linguistic knowledge — knowledge about the meanings of utterances and plans for expressing meanings as text.
- Discourse knowledge — knowledge about the intentional, social, and rhetorical relationships that link utterances.
- Domain knowledge — factual knowledge of the medical domain and the medical case that is under consideration.
- Pedagogy — knowledge about the tutoring task.
- Decision-support — knowledge about the statistical model and how to interpret the information that is derivable from the model.

In B2, the interaction between the tasks is possible because the information for all knowledge sources is represented in a uniform framework.

The primary advantage of a uniform representation is that it eliminates knowledge interchange overhead. That is, there are no special-purpose reasoners with specialized knowledge representation(s), and all reasoning uses the same reasoner. We believe that this may scale better than the traditional, non-uniform approach. We are not alone in advocating a uniform representation, see for example Soar (Rosenbloom, Laird, & Newell 1993).

The traditional approach to building intelligent, interactive systems is to "compartmentalize" the special-purpose reasoners with different knowledge representations appropriate to the specialized tasks. This is efficient in the initial stages of system building, however as a system matures, components with rich, detailed representations will have to communicate with components having more superficial representations. Knowledge interchange is a serious problem, even in systems that have a common knowledge representation ancestor, such as KL-ONE (Heinsohn *et al.* 1994). One common problem that arises is conflicting ontologies (Traum *et al.* 1996). For example the TRAINS-93 system has many special-purpose components where each component has its own fairly sophisticated representation (Logical Form, Episodic Logic, Conversation Representation Theory, Tyro, Event-based Temporal Logic). In later work with the TRAINS-96 system there is still the stratified architecture, however the components all have more superficial representations, and communicate with each other in KQML (Ferguson *et al.* 1996).

Uniform representations have not been used in traditional intelligent interfaces for reasons of perceived computational and management complexity. Management complexity can be addressed by the use of standards that enforce the goal of uniformity. Computational complexity is more problematic, as the speed of inference in a monolithic knowledge base has been shown to grow in proportion to the knowledge (Heinsohn *et al.* 1994). Computational complexity can be addressed either by distributing computation for reasoning and knowledge base access (Geller 1994) or by structuring knowledge for efficient access, for example by adding meta-knowledge that specifies the nature of the knowledge in the uniform knowledge base. In our work, we use meta-facts (*e.g.* to say that a given fact is about the user model) to index the knowledge base, so that reasoning about the user model can be done without search.

We also attempt to limit the growth of the knowledge base by using so-called "mixed-depth representations" of expressions and actions. A *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 & Ryan 1992). Moreover, "shallow" and "deep" are a matter of degree. 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. Unlike quasi-logical form, which is used primarily for storage of information, mixed-depth representations are well-formed propositions, subject to logical inference. Disambiguation and interpretation, when it occurs, is done by reasoning.

A General Architecture for Dialog

Our architecture for Intelligent Dialog Systems is shown in Figure 2. The INPUT MANAGER and DISPLAY MANAGER deal with input and output, respectively. The input modalities would include typed text, spoken text, mouse clicks, and drawing. The output modalities would 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

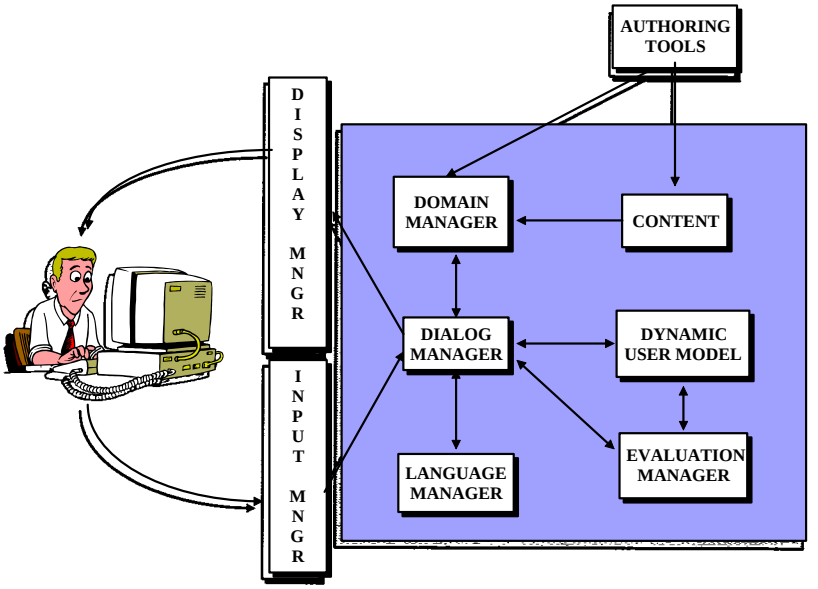

Figure 2: Architecture of an Intelligent Dialog System

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.

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. 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.

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 a mixed-depth representation of the user's actions. This representation includes a syntactic analysis and a partial semantic analysis. 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 , we will illustrate our approach by working through the answer to a question: *What is Mary's age?*.

An Example

Computationally, the system processes dialog by parsing communicative acts into mixed-depth representations, 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 to process 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 achievement involves a plan that 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.

Parsing, content, and anaphora interpretation

As previously mentioned, the question is parsed using a broad-coverage grammar which builds the mixed-depth representation(s) as shown in Figure 3. For clarity, the representations are shown as simplified feature structures. Propositions are labeled as M*j* and (potential) discourse entities are labeled as B*k*. In Figure 3, 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 an 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`).

As can be seen from Figure 3, the propositions produced by the parser are the weakest possible interpretations of the utterance. Any question of this form would parse into similar propositions; their 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 4 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 5.[1] In addition, a separate rule (not shown) will be used to establish an equivalence relationship between B3 (the Mary mentioned in the utterance) and B0 (the Mary known to the system).[2] As a result of the rule in Figure 4, 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 4 and 5, variables are labeled as V*n* and, for clarity, the bindings of the variables of the rules are shown relative to the original question in the lower right corner of Figure 4. The `if` part of the rule in Figure 4, 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 4 allows the interpretation of the mixed-depth representations of Figure 3 as a proposition, which expressed in a logical formula, is *has-property*(`Mary, age, what`)

Pragmatic interpretation

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 6 and 7 that the system uses to derive a possible interpretation.

[1] Elements 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.

[2] Currently, the system assumes that all objects with the same name are the same entity.

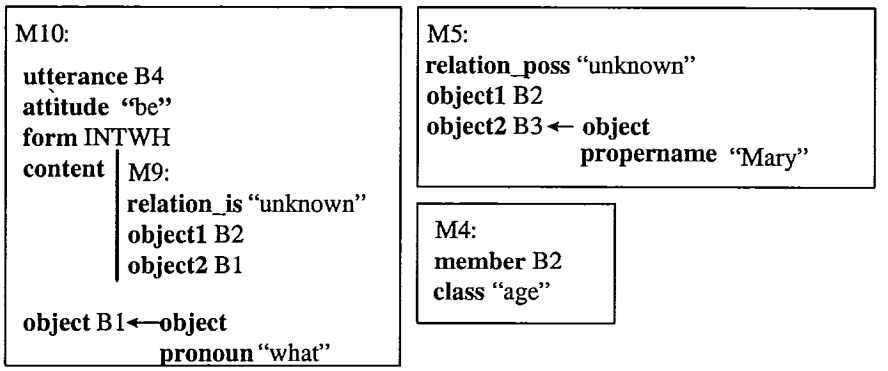

Figure 3: Initial Mixed-Depth Representation of: *What is Mary's age?*

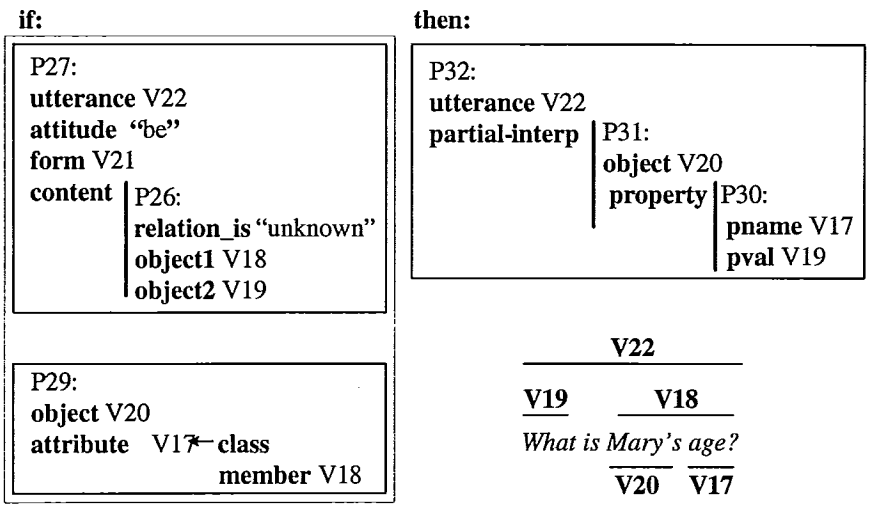

Figure 4: Partial Interpretation Rule for the Utterance B4

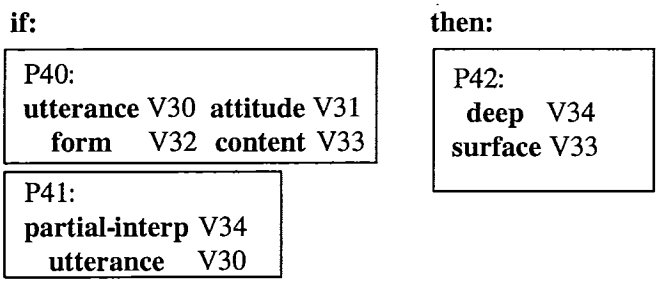

Figure 5: Derivation Rule for Making Explicit the Relation between an Utterance's Content and Partial Interpretation

The rule in Figure 6 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`[3] by performing the (linguistic) action

[3] An `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.

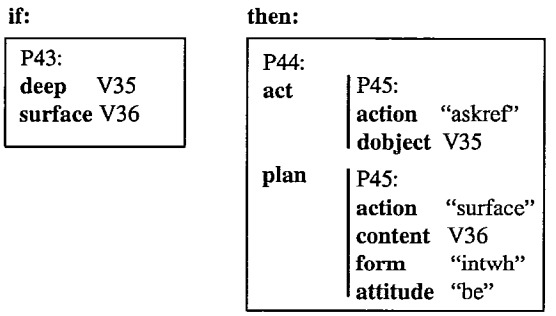

Figure 6: Text Planning Rule

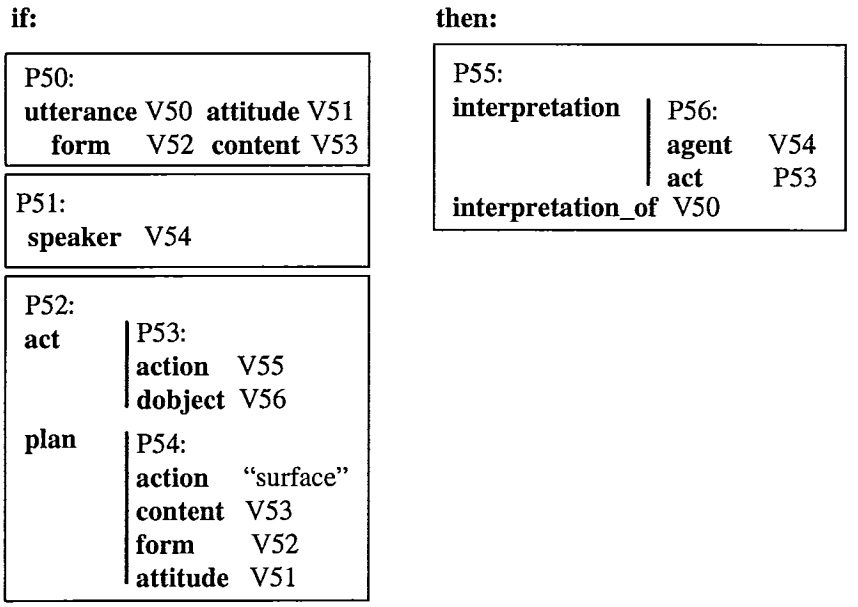

Figure 7: Pragmatic Interpretation Rule to Identify Communicative Act

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 7 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))
```

The final interpretation of the original utterance B4 is shown in Figure 8. M22 is the interpretation, namely that the user is performing an `askref` on *what is Mary's age* (M24) and the `system`. More concisely, the system has interpreted the original utterance *what is Mary's age* as the user asking the system: *what is Mary's age?*.

At this point our discussion of interpretation is complete (we will not consider, here, the possibility of misunderstandings or arguments). Next, we will consider response generation, as it illustrates the link between inference and action in the underlying

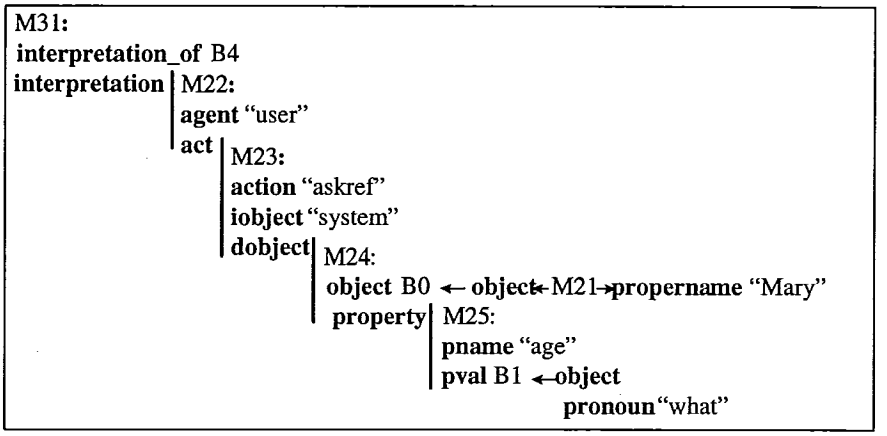

Figure 8: An Interpretation of the Utterance *What is Mary's age?*

knowledge base.

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 9 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.[4] 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).

Summary

This research supports robust, flexible, 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 model 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 discourse model captures 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.

[4] Compatibility is a notion that is related to the coherence of dialog and the expression of reflexive intentions (following (McRoy & 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.

References

Bordegoni, M.; G.Faconti; Maybury, M. T.; Rist, T.; Ruggieri, S.; Trahanias, P.; and Wilson, M. 1997. A standard reference model for intelligent multimedia presentation systems. In *Proceedings of the IJCAI '97 Workshop on Intelligent Multimodal Systems.*

Ferguson, G. M.; Allen, J. F.; Miller, B. W.; and Ringger, E. K. 1996. The design and implementation of the trains-96 system: A prototype mixed-initiative planning assistant. TRAINS TN 96-5, Computer Science Dept., University of Rochester.

Geller, J. 1994. Advanced update operations in massively parallel knowledge representation. In Kitano, H., and Hendler, J., eds., *Massively Parallel Artificial Intelligence.* MIT Press. 74–101.

Grosz, B. J., and Sidner, C. L. 1986. Attention, intentions, and the structure of discourse. *Computational Linquistics* 12.

Haddawy, P.; Jacobson, J.; and Jr., C. E. K. 1997. Banter: A bayesian network tutoring shell. *Artificial Intelligence and Medicine* 10(2):177–200.

Heinsohn, J.; Kudenko, D.; Nebel, B.; and Profitlich, H.-J. 1994. An empirical analysis of terminological representation systems. *Artificial Intelligence* 68:367–397.

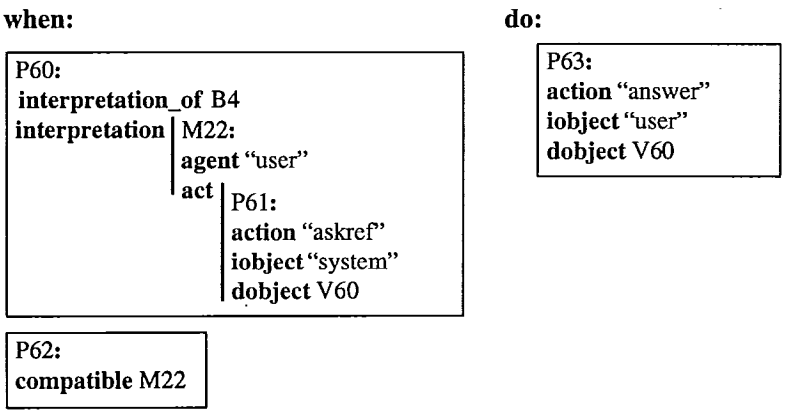

Figure 9: An Acting Rule for Responding to a Question

Hirst, G., and Ryan, M. 1992. Mixed-depth representations for natural language text. In Jacobs, P., ed., *Text-Based Intelligent Systems.* Lawrence Erlbaum Associates.

Lambert, L., and Carberry, S. 1991. A tri-partite plan-based model of dialogue. In *29th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 47–54.

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

McRoy, S. W.; Channarukul, S.; and Ali, S. S. 2000. Text realization for dialog. In *This volume.*

Moore, J., and Paris, C. 1993. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics* 19(4):651–695.

Restificar, A. C.; Ali, S. S.; and McRoy, S. W. 1999a. ARGUER: Using Argument Schemas for Argument Detection and Rebuttal in Dialogs. In *Proceedings of User Modeling 1999.* Kluwer.

Restificar, A. C.; Ali, S. S.; and McRoy, S. W. 1999b. Argument Detection and Rebuttal in Dialog. In *Proceedings of Cognitive Science 1999.*

Rosenbloom, P.; Laird, J.; and Newell, A., eds. 1993. *The Soar Papers: Readings on Integrated Intelligence.* MIT Press.

Shapiro, S. C., and Rapaport, W. J. 1992. The SNePS family. *Computers & Mathematics with Applications* 23(2–5).

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