12

ARPA ORDER NO. 2930
NR 154-374

ISI/RR-77-52
January 1977

James A. Moore
William C. Mann
James A. Levin

A Goal-Oriented Model of Natural Language Interaction

INFORMATION SCIENCES INSTITUTE
UNIVERSITY OF SOUTHERN CALIFORNIA ISI
4676 Admiralty Way/Marina del Rey/California 90291
(213) 822-1511

The research reported herein is supported by the Personnel and Training Research Programs of the Office of Naval Research (Code 458), Arlington VA 22217, Contract Authority Identification Number NR 154-374, Contract N00014-75-C-0710, under ARPA Order Number 2930 from the Cybernetics Technology Office of the Advanced Research Projects Agency. The report is approved for public release; distribution is unlimited. Reproduction in whole or in part is permitted for any purpose of the United States Government. Opinions expressed herein are those of the authors and do not necessarily reflect policy of any agency of the United States Government.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

14 1. ~~REPORT NUMBER~~
ISI/RR-77-52

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

6 4. TITLE (and Subtitle)
A GOAL-ORIENTED MODEL OF NATURAL LANGUAGE INTERACTION.

9 5. ~~TYPE OF REPORT & PERIOD COVERED~~
RESEARCH rept.,

6. ~~PERFORMING ORG. REPORT NUMBER~~

10 7. AUTHOR(s)
James A. Moore, James A. Levin, William C. Mann

15 8. ~~CONTRACT OR GRANT NUMBER(s)~~
N00014-75-C-0710,
ARPA Order-2930

9. ~~PERFORMING ORGANIZATION NAME AND ADDRESS~~
USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, California 90291

10. ~~PROGRAM ELEMENT, PROJECT, TASK~~ AREA & WORK UNIT NUMBERS
61153N RR042-06-01
RR042-06 NR154-374

11. CONTROLLING OFFICE NAME AND ADDRESS
Cybernetics Technology Office
Advanced Research Projects Agency
1400 Wilson Blvd., Arlington, VA 22209

11 12. ~~REPORT DATE~~
Jan[redacted] [redacted]77

13. ~~NUMBER OF PAGES~~
61

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
Personnel & Training Research Programs
Office of Naval Research - Code 458
800 No. Quincy Street
Arlington, VA 22217

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

12 59p. 16 RR04206

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

17 RR0420601

18. SUPPLEMENTARY NOTES

To be submitted to the American Journal of Computational Linguistics

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

artificial intelligence, cognitive psychology, complexity, computer, computational linguistics, dialogue, evaluation, knowledge structures, linguistic, model, performance, problem solving, process, research methodology, theory.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

(over)

407 952

1/B

20. ABSTRACT

This report summarizes a research program in modeling human communicative ability. The methodology involved selecting a single, naturally-occurring dialogue, instructing a human observer to extract certain aspects of the dialogue relating to its comprehension, and then using these aspects to guide the model building and verification activity.

Behind the model is the development of an innovative theory of language, according to which people engage in language behavior in order to pursue their own goals. Dialogues communicate effectively because of a shared understanding of a collection of interrelated, cooperative goal structures, held and pursued by the language users. The recognition of this shared knowledge within the model contributed to an initial accounting of implicit communication.

The report contains a detailed statement of the problem, a review of the state of related research, a description of the contributions of this research to linguistic theory, and an explanation of the research methodology. It then describes the model in its current state of development and discusses a detailed simulation of it. Next the role of the Observer in the research is explored with emphasis on the nature of his task.

A GOAL-ORIENTED MODEL OF NATURAL LANGUAGE INTERACTION

James A. Moore
James A. Levin
William C. Mann

ISI/RR-77-52

January 1977

USC/Information Sciences Institute
4676 Admiralty Way
Marina Del Rey, California 90291

The research reported herein is supported by the Personnel and Training Research Programs of the Office of Naval Research (Code 458), Arlington VA 22217, Contract Authority Identification Number NR 154-374, Contract N00014-75-C-0710, under ARPA Order Number 2930 from the Cybernetics Technology Office of the Advanced Research Projects Agency. The report is approved for public release; distribution is unlimited. Reproduction in whole or in part is permitted for any purpose of the United States Government. Opinions expressed herein are those of the authors and do not necessarily reflect policy of any agency of the United States Government.

CONTENTS

ABSTRACT

This report describes a research program in modeling human communication. The methodology involved selecting a single, naturally-occurring dialogue, instructing a human observer to extract certain aspects of the dialogue relating to its comprehension, and then using these aspects to guide the building and verification of a model of the dialogue participants.

Underlying the model is the development of a new theory of language, according to which people engage in language behavior in order to pursue their own goals. People are able to communicate effectively in dialogues because they share an understanding of a collection of interrelated, cooperative goal structures, held and pursued by each other. The recognition of this shared knowledge within the model leads to an accounting of some aspects of implicit communication.

This report contains a detailed statement of the problem, a review of related research, and a description of the contributions of this research to linguistic theory. It then describes the dialogue model in its current state of development (with a detailed simulation of the model in the Appendix) followed by an exploration of the methodology of which this model development is a part. Finally, the deficiencies of existing man-machine interfaces are summarized, in the light of this research.

STATEMENT OF THE PROBLEM

Our broadest goal is to improve the sorry state of interactive man-machine communication, including its appearance of complexity, rigidity, lack of continuity and the difficulty many people have in acquiring useful levels of competence. In our pursuit of this, we have found it appropriate to adopt the following two assumptions:

> Assumption 1: When people communicate with machines, they do so by adapting their already well-developed ability to communicate with other people.
>
> Assumption 2: The effectiveness of this communication is diminished by any kind of required adaptation.

A scientific understanding of how people communicate is thus relevant to the design of man-machine communication schemes, but such knowledge is seldom used in the design process. Human communication skills have not been characterized at a level of detail appropriate for guiding design, and so interface designers cannot take into account some major determinants of their success.

The principal goal of this research was therefore to ***create a model of human communication*** at an appropriate level of detail to benefit man-machine communication design.

It is evident that any form of communication must be based on a collection of knowledge shared by the individuals engaged in that communication. However, the nature of this shared knowledge and how it is used in the communicative process are less evident. We have developed a working hypothesis which has deeply affected the research:

> Hypothesis: People share knowledge of what kinds of goals may be pursued by communicating, and how communication tends to satisfy these goals; they use this knowledge as an essential component of their comprehension and generation of natural language.

In particular, the act of generating language is performed in a manner which tends to advance the goals of the person generating it; thus, part of interpreting a particular use of language is the attempt to identify the goals of the person generating it.

Working with this hypothesis, we have engaged in three related investigations:

1. Study naturally occurring language to discover regularities of usage and to determine how these regularities contribute to the communicative aspects of language.

2. Represent the understanding of these regularities as data structures and process models.

3. Establish standards by which their performance can be compared with that of humans on closely related tasks.

We have adopted a number of additional temporary selectivity constraints on the task:

1. We are only modelling the ***receptive*** aspects of communication.

2. The only type of communication we are examining is ***dialogue***, interaction in real-time, by exactly two people.

3. This dialogue is conducted over a ***restricted medium*** so that there is no visual or intonational communication, which would not be captured in the transcript.

4. No attempt is made to design the model to operate in time commensurate with the dialogue itself.

PAST RESEARCH ON LANGUAGE COMPREHENSION

Most of the research into language comprehension has focused on the comprehension of single sentences or fragments of sentences. However some research has indicated the importance of the context created by surrounding sentences on the comprehension of an individual sentence. Much of this research has studied the comprehension of stories, starting with Bartlett (1932), who found that stories influenced the ability of subjects to recall the individual utterances within that story. In particular, he found that some sentences that did not make sense within the rest of the story were replaced in the recalls by other sentences that were similar in some ways, but differed so that they fit the story.

A similar result was found by Bransford and Johnson (1973), using "ambiguous stories". They generated stories, each of which could be interpreted in two widely different ways, and influenced the interpretation derived by subjects by giving each story one of two titles. For example, one story was titled either "Watching a peace march from the fortieth floor" or "A space trip to an inhabited planet". Most of the sentences in the story could be interpreted either way, but one sentence made sense only within one of these two interpretations. Subjects given one title were able to recall this sentence well, but those given the other title (with the incompatible interpretation) were not. Generally, these results indicate that knowledge spanning multiple sentences is involved in comprehending each individual sentence of a story. This multi-sentential knowledge is used to tie the comprehension of each sentence together, and any sentence which does not fit into this knowledge is not easily assimilated or remembered.

A specific model for the form of this multi-sentential knowledge is the "story schema", organized within a story grammar (Rumelhart, 1975). This model has been supported by the results of story recalls (Rumelhart, 1975; Thorndyke, 1977). Other similar kinds of theoretical constructs for organizing multiple sentences of stories have been proposed called: "frames" (Minsky, 1975; Charniak, 1975), "scripts" (Schank & Abelson, 1975), and "commonsense algorithms" (Rieger, 1975).

To account for the conduct and comprehension of dialogues, multi-sentential knowledge units have also been proposed by linguists and sociolinguists to explain certain kinds of regularities observed in naturally occurring dialogues. These regularities have been called "rules" by Labov & Fanshel (1974) and "sequences" by Sacks, Schegloff, & Jefferson (1974).

Once these multi-sentential knowledge units are evoked, they serve as a basis for comprehending the successive inputs. This is achieved by generating expectations and by providing a framework for integrating the comprehension of an utterance with that of its predecessors. Recently, we proposed (Levin & Moore, 1976) that multi-sentential knowledge units are specified primarily by the speaker's and hearer's goals. This differs from the other proposed multi-sentential units, some of which are specified only by co-occurrence properties, others by causal characteristics. These goal-oriented units, which we call Dialogue-games* (Levin & Moore, 1976), specify the kinds of language interactions in which people engage, rather than the specific content of these interactions.

People use language primarily to communicate with other people to achieve their own goals. The Dialogue-game multi-sentential structures were developed to represent this knowledge about language and how it can be used to achieve goals.

An important problem facing researchers in language comprehension is posed by sentences with which the speaker performs what philosophers of language have called "indirect speech acts" (Searle, 1969). The direct comprehension of these sentences fails to derive the main communicative effect. For example, declarative sentences can be used to seek information ("I need to know your social security number."); questions can be used to convey information ("Did you know that John and Harriet got married?") or to request an action ("Could you pass the salt?"). These kinds of utterances, which have been extensively analyzed by philosophers of language (Austin, 1962; Searle, 1969, 1975; Grice, 1975), are not handled satisfactorily by any of the current theories of the direct comprehension of language. However, these indirect language usages are widespread in naturally occurring language--even two year old children can comprehend indirect requests for action almost as well as direct requests (Shatz, 1975).

One theory proposed to account for these indirect uses of language is based on the concept of "conversational postulates" (Grice, 1975; Gordon & Lakoff, 1971). If the direct comprehension of an utterance is implausible, then the indirect meaning is derived using these postulates. Clark & Lucy (1975) formalized and tested this model, and found support for a three stage model (deriving the literal meaning, check its plausibility, and if implausible, deriving the "intended" meaning" from conversational rules).

In general, this approach to indirect speech acts is inference-based, depending on the application of conversational rules to infer the indirect meaning from the direct meaning and the context. A different approach has been proposed by Labov & Fanshel (1974) and by Levin & Moore (1976). Multi-sentential knowledge, organizing a segment of language interaction, can form the basis for deriving the indirect effect of utterance within the segment. For example, a multi-sentential structure for an information-seeking interaction can supply the appropriate context for interpreting the subsequent utterances to seek and then supply information. The inference-based approach requires one set of conversational rules for information requests, a different set of rules for answers to these requests, and a way to tie these two rule sets together. The Dialogue-game model postulates that there is but one knowledge structure for this kind of interaction, and leads to a model of three sets of cooperating processes: (l) processes for recognizing when this kind of interaction is proposed, (2) processes for using this knowledge to comprehend utterances within its scope, and (3) processes for identifying when the interaction is to be terminated.

* The term "Dialogue-game" was adopted by analogy from Wittgenstein's term "language game" (Wittgenstein, 1958). However, Dialogue-games represent knowledge people have about language as used to pursue goals, rather than Wittgenstein's more comprehensive notion. Although there are also similarities with other "games," the properties of Dialogue- games are only those described here. For example, they are not necessarily competitive, consciously pursued, or zero-sum.

THE SHAPE OF THE THEORY

Our theory of human use of language has been strongly influenced by work in human problem solving, (Newell & Simon, 1972) in which the behavior of a human is modeled as an information-processing system, having goals to pursue and selecting actions which tend to achieve the goals.

We view humans as engaging in linguistic behavior in order to advance the state of certain of their goals. They decide to use language, they select (or accept) the other participant for a dialogue, they choose the details of linguistic expression -- all with the expectation that some of their desired state specifications can thereby be realized. Furthermore, they break off an interaction either when the relevant goals have been satisfied, or when it becomes clear that they cannot be. In this theory of language, a participant in a linguistic exchange views the other as an independent information processing system, with separate knowledge, goals, abilities and access to the world. A speaker has a range of potential changes he can effect in his listener, a corresponding collection of linguistic actions which may result in each such change, and some notion of the consequences of performing each of these. The speaker may view the hearer as a resource for information, a potential actor, or as an object to be molded into some desired state.

A dialogue involves two speakers, who alternate as hearers. In choosing to initiate or continue the exchange, a participant is attempting to satisfy his own goals; in interpreting an utterance of his partner, each participant attempts to find the way in which that utterance serves the goals of his partner. Thus a dialogue continues because the participants can continue to see it as furthering their own goals. Likewise, when the dialogue no longer serves the goals of one of the participants, it is redirected to new goals or terminated.

This mechanism of joint interaction, via exchange of utterances, in pursuit of desired states, is useful for achieving certain related pairs of participants' goals (e.g., learning/teaching, buying/selling, getting help/giving help, ...). For many of these goal-pairs there are highly structured collections of knowledge, shared by the members of the language community. These collections specify such things as: 1) what characteristics an individual must have to engage in a dialogue of this sort, 2) how this dialogue is initiated, pursued and terminated, 3) what range of information can be communicated implicitly, and 4) under what circumstances the dialogue will "succeed" (serve the function for which it was initiated) and how this will be exhibited in the participants' behavior.

In the Dialogue Modeling System, we have attempted to discover and represent these collections of knowledge and model the way in which they are used to facilitate the comprehension of a dialogue.

THE DIALOGUE-GAME MODEL

This section describes our Dialogue-games model at its current state of development. It starts with a brief overview of dialogue and how it is regulated, then describes the dominant knowledge structures which gu·· the model, and finally describes a set of processes which apply these knowledge structures to text, comprehending it.

Each participant in dialogue is simply pursuing his own goals of the moment. The two participants interact smoothly because the conventions of communication coordinate their goals and give them continuing reasons to speak and listen. These goals have a number of attributes which are not necessary either to human activity nor to communication, but which are characteristic of a wide range of dialogue types:

1. ***They are cooperatively established.*** There are bidding and acceptance activities that serve to introduce goals.

2. ***They are mutually known.*** Each party assumes or comes to know goals of the other, and each interprets the entire dialogue relative to currently known goals.

3. ***They are configured by convention.*** Sets of goals for use in dialogue (and other language use as well) are tacitly known and employed by all competent speakers of the language.

4. ***They are bilateral.*** Each dialogue participant assumes goals complementary to those of his partner.

An uninterrupted dialogue goes through three phases:

- establishing goals,
- pursuing goals,
- decommitting from goals.

Typically this sequence is compounded and repeated several times in the course of a few minutes.

We have created knowledge structures to represent these conventions, and processes to apply the conventions to actual dialogues to comprehend them. Since the knowledge structures dominate all of the activity, they are described first.

Text is interpreted in this model by frequent modification of a "Workspace"* which represents the attention or awareness of the listening party. The modifications are roughly cyclic:

1. A new item of text T is brought into attention through the "Parser."*

2. Interpretive consequences of T are developed in the Workspace by a variety of processes.

3. An expression E appears in the Workspace which relates T to the imputed goals of the speaker of T.

This final expression is of course a formal expression in the knowledge representation of the model. E represents the proposition (held by the hearer) that in uttering T, the speaker was performing an act in pursuit of G, a speaker's goal known to the hearer. Comprehension is equated with relating text to satisfaction of speaker's goals.

To make an explicit account of dialogue in this way, we now describe the knowledge structures which represent those conventions which supply the goals for the participants to pursue. In particular, we will answer the following three questions:

1. What is the knowledge we are representing within the definition of a particular Dialogue-game?

2. How is this knowledge used to model the receptive acts of dialogue participants?

3. What sort of processes does it take to support this model?

What's in a Game?

A Dialogue-game consists of three parts: a set of ***Parameters***, the collection of ***Specifications*** that apply to these Parameters throughout the conduct of the game, and a partially ordered set of ***Components*** characterizing the dynamic aspects of the game.

For the balance of this section, we will elaborate on these three parts and exemplify these with an example of the Helping-game.

Bidding and Acceptance are entry operations which people use to enter Dialogue-games. Bidding:

1. identifies the game,
2. indicates the bidder's interest in pursuing the game, and
3. identifies the Parameter configuration intended.

Bidding is performed many different ways, often very briefly. It is typically the source of a great deal of implicit communication, since a brief bid can communicate all of the Parameters and their Specifications for the Dialogue-game being bid.

---

* The Parser and the Workspace are parts of the process model and are described in a later section.

Acceptance is one of the typical responses to a Bid, and leads to pursuit of the game. Acceptance exhibits:

1. acknowledgement lhat a bid has been made,
2. recognition of the particular Dialogue-game and Parameters bid,
3. agreement to pursue the game,

and

4. assumption of the Acceptors role in the Dialogue-game.

Acceptance is often implicit, especially in relatively informal dialogue. It can be indicated by statements of agreement or approval, or by beginning to pursue the game (i.e. attempts to satisfy the goals). Alternatives to acceptance include rejection, negotiation and ignoring.

Bidding and acceptance appear to be part of game entry for all of the dialogue games of ordinary adult dialogue. They are also involved in game termination. In the case of termination, there are three other alternatives: interruption and spontaneous termination by either goal satisfaction or unconditional goal failure.

Parameters

Dialogue-games capture a certain collection of information, common across many dialogues. However, the individual participants involved, and the subject (but not the function) of the dialogue may vary freely over dialogues described by the same Dialogue-game. To represent this, each Dialogue-game has a set of Paramelers which assume specific values for each particular dialogue.

The dialogue types we have represented so far as Dialogue-games have required only these three Parameters: the two participants involved (called "Roles"), and the subject of the dialogue (called "Topic").

Parameter Specifications

One of the major aspects distinguishing various types of dialogues is the set of goals held by the participants. Another such aspect is the individual knowledge states of the participants. We have found that for each type of dialogue, there is a corresponding set of descriptions which must hold for the goal and knowledge states of the participants, vis-a-vis each other and the subject. Within the formalism of the Dialogue-game, these are called the Parameter Specifications, and are represented by a collection of predicates on the Parameters.

We claim that these Specifications are known to the participants of the dialogue, and the requirement that they be satisfied during the conduct of a game is used by the participants to: signal what game(s) they wish to conduct, recognize what game is being bid, decide how to respond to a bid, conduct the game once the bid is accepted and terminate the game when appropriate. These Specifications also provide the means with which to explain the implicit, but clearly successful, communication which accompanies any natural dialogue.

Examples and discussions of thoso Specifications will accompany the example of the Helping-game, below.

Components

The parameter Specifications represent those aspects of a dialogue type that remain constant throughout the course of a dialogue of that type. We have also found that certain aspects change in systematic ways; these are represented in Dialogue-games as Components. In the Dialogue-games we have developed so far, the Components have been represented as a set of participants' subgoals, partially ordered in time.

Once a game has been, in effect, bid and accepted, the two participants each pursue the subgoals specified for their role in the Components of this game. These subgoals are mutually complementary -- each set facilitating the other. Furthermore, b/ the time the termination stage has been reached (subject to a few constraints) pursuit of the Component-specified subgoals will have assured satisfaction of the higher, initial goals of the participants, in service of which the game was initiated in the first place.

The Helping-game, an Example

In this section, we exhibit a specific Dialogue-game: the *Helping-game*. This game is presented in an informal representation, in order to emphasize the informational content, rather than the representational power of our formalism. Later in this report we will present the formal analogue of this same game. In what follows, the italics indicates the information contained in the representation of this particular Dialogue-game; the intervening text is explanatory commentary.

The (annotated) Helping-game

There are three Parameters: HELPEE, HELPER, and TASK.

The HELPEE wants help from the HELPEE. The TASK is some sort of a problem, otherwise unspecified.

The parameter Specifications are as follows:

The HELPEE is a person.

The HELPEE wants to perform the TASK.

The HELPEE wants to be able to perform the TASK.

The HELPEE is not able to perform the TASK.

The HELPEE is permitted to perform the TASK.

These Specifications not only constrain who would qualify as filling the role of HELPEE, but also provide reliable information about the HELPEE, given that this individual is believed to be engaged in the Helping-game. This prohibits, for example, someone from asking for help (sincerely) on a problem he did not want to solve. Similarly, if one receives what he judges to be a sincere request for help to do something, he normally assumes that the requester has the necessary authority to do it, if only he could.

The HELPER is a person.

The HELPER wants to help the HELPEE perform his TASK.

The HELPER is able to provide this help.

So, in order to be a HELPER, an individual must be willing and able to provide the needed assistance. Since this Dialogue-game represents *shared* knowledge, the HELPER knows these Specifications, and therefore will not bid the Helping-game to someone who is not likely to meet them. And similarly, no one who fails to meet these Specifications (and knows he fails) will accept a bid for the Helping-game with himself as HELPER.

The Components of the Helping-game are the following:

(Diagnosis phase -- communicate what the problem is.)

HELPEE wants the HELPER to know that a sequence of unexceptional events happened.

The HELPEE sets up a context by describing a world where everything, so far, is going well. Since the situation (involving HELPEE attempting to do the TASK) is presumed to be known by the HELPER, it is further assumed that the HELPER's expectations for subsequent activity will closely parallel those of the HELPEE.

Then, the HELPEE wants the HELPER to know about one or more events which either:
1) happened and were not expected
or
2) did not happen and were expected.

This very frequent pattern of conducting a Helping-game is sufficiently well ingrained in the participants, that the HELPEE almost never needs to actually ask a question at this point. By simply exhibiting a failure of expectation, the HELPEE has communicated that this acts as a block to his successfully pursuing the TASK. Furthermore, he expects the HELPER to explain why this failure occurred and how he can avoid it or otherwise continue to perform the TASK.

(Treatment phase -- communicate the explanation for the perceived failure.)

Then, the HELPER wants the HELPEE to know about an action which, when performed by the HELPEE, will enable him to pursue the TASK.

The context description has enabled the HELPEE to identify a collection of activities which he understands, and in which the HELPEE is attempting to participate. The violation-of-expectation description points out just where the HELPEE's image of the activities differs from the HELPER's (presumably correct) image. It is from this discovered area of difference that the HELPER selects an action for the HELPEE which is expected to solve his problem.

Dialogue-games in the Comprehension of Dialogue

In this section we describe the five stages of dialogue assimilation and detail the involvement of Dialogue-games with each stage:

1) nomination,
2) recognition,
3) instantiation,
4) conduct,
and
5) termination.

Processing Environment

Our description of the model should be viewed as representing the changing cognitive state of *one* of the participants, throughout the course of the dialogue. That is, there are actually two models involved, one for each participant. Since the processing afforded each is the same, we will describe only one.

The Dialogue Modeling System consists of a long-term memory (LTM), a workspace (WS), and a set of processes that modify the contents of WS, contingent upon the contents of LTM and WS. LTM contains a representation of the knowledge that the particular dialogue participant being modeled brings to the dialogue before it starts. This includes knowledge about the world, relevant objects, processes, concepts, the cognitive state of his partner in dialogue, rules of inference and evidence, as well as linguistic knowledge: words and their semantic representation, case frames for verbs and predicates and, of course, the multi-turn language structures, the Dialogue-games.

WS is the volatile short-term memory of the model, containing all the partial and temporary results of processing. The contents of WS at any moment represent the model's state of comprehension and focus at that point. The processes are autonomous specialists, operating independently and in parallel, to modify the entities in WS (called "activations"). These processes are also influenced by the contents of WS, as well as by the knowledge in LTM. Thus, WS is the place in which these concurrently operating processes interact with each other. This anarchistic control structure resembles that the HEARSAY system (Erman, et. al., 1973)

Nomination

When dialogue participants propose a new type of interaction, they do not consistently use any single word or phrase to name the interaction. Thus we cannot determine which Dialogue-game(s) represent the dialogue type, through a simple invocation by name (or any other pre-known collection of words or phrases). Instead the dialogue type is communicated by attempts to establish various entities as the values of the

Parameters of the desired Dialogue-game. Thus, an utterance which is comprehended as associating an entity (a person or a concept) with a Parameter of a Dialogue-game suggests that Dialogue-game as a possibility for initiation.

The Dialogue-game Model has two ways in which these nominations of new Dialogue-games occur. One of the processes of the model is a "spreading activation" process called Proteus (Levin, 1976). Proteus generates new activations in WS on the basis of relations in LTM, from concepts that are already in WS. Proteus brings into focus concepts somehow related to those already there. A collection of concepts in WS leads to focusing on some aspect of a particular Dialogue-game, in this sense "nominating" it as a possible new Dialogue-game.

MATCH and DEDUCE are two of the model's processes which operate in conjunction to generate new activations from existing ones, by means of finding and applying rule-like transformations. They operate through partial match and plausible inference techniques, and if they activate Parameters, then the Dialogue-game that contains those Parameters becomes nominated as a candidate Dialogue-game. Match and Deduce operate together as a kind of production system (c.f, Newell, 1973).

For example, from the input utterance:

"I tried to send a message to P at S and it didn't go."

the following two sequences of associations and inferences result:
(1a) I tried to X.
(2a) I wanted to X.
(3a) I want to X.
(4a) HELPEE wants to do TASK.

(1b) It didn't go.
(2b) What I tried to do didn't work.
(3b) X didn't work.
(4b) I can't X.
(5b) I don't know how to X.
(6b) HELPEE doesn't know how to do TASK.

(Where: I = HELPEE and X = do TASK = send a message to P at S.)

At this point, (4a) and (6b), since they are both Parameter Specifications for the Helping-game, cause the model to focus on this Dialogue-game, in effect nominating it as an organizing structure for the dialogue being initiated.

Recognition

The processes described so far are reasonably unselective and may activate a number of possible Dialogue-games, some of which may be mutually incompatible or otherwise inappropriate. There is a process called the Dialogue-game Process, which investigates each of the nominated Dialogue-games, verifying inferences based of the parameter Specifications, and eliminating those Dialogue-games for which one or more Specifications are contradicted.

A second mechanism (part of Proteus) identifios those activations which are incompatible and sels about accumulating evidence in support of a decision lo accept one and delete the rest from the WS.

For example, suppose the utterance:

How do I get RUNOFF to work?

leads to the nomination of two games:

Helping-game (person asking question wants to know answer)
and
Info-probe-game (person asking question wants to know if other knows answer)

These two Dialogue-games have a lot in common but differ in one crucial aspect: In the Helping-game, the questioner does not know the answer to the question, while in the Info-probe-game he does. These two predicates are represented in the Parameter Specifications of the two Dialogue-games, and upon their joint nomination are discovered to be contradictory. Proteus represents this discovery with a structure which has the effect of extinguishing the conflicting Dialogue-game for which lhere is the least supporting evidence. Such support might be, for example, either the knowledge that the speaker is the hearer's teacher or that he is a novice programmer (which would lend support for the choice of the Info-probe-game or Helping-game, respectively).

Through these processes, the number of candidate Dialogue-games is reduced until those remainirg are compatible with each other and the knowledge currently in WS and in LTM.

Instantiation

Once a proposed Dialogue-game has successfully survived the filtering processes described above, it is then instantiated by the Dialogue-game Process. Those parameter Specifications not previously known (represented in the WS) are established as new inferred knowledge about the Parameters. It is through these instanliation processes that a large part of the implicit communication between participants of the dialogue is modeled.

To illustrate this, suppose that the following are represented in WS (i.e., known):

SPEAKER does not know how to do a TASK.
SPEAKER wants to know how to do that TASK.
SPEAKER wants to do the TASK.

These are adequate to nominate the Helping-game. In the process of inslantialing lhis Dialogue-game, the following predicates are added to WS:

SPEAKER believes HEARER knows how to do TASK.
SPEAKER believes HEARER is able lo tell him how to do TASK.
SPEAKER believes HEARER is willing to tell him how to do TASK.
SPEAKER wants HEARER to tell him how to do TASK.
SPEAKER expects HEARER to tell him hnw to do TASK.

The model, then, predicts that these predicates are implicitly communicated by an utterance which succeeds in instantiating the Helping-game. This corresponds to a dialogue in which "I can't get this thing to work" is taken to be a request for help (which it clearly is not, on the surface).

Conduct

Once a Dialogue-game is instantiated, the Dialogue-games Process is guided by the Components, in comprehending the rest of the dialogue. These Components are goals for the dialogue participants. For the speaker, these goals guide what he is next to say; for the hearer, these provide expectations for the functions to be served by the speaker's subsequent utterances.

As we will see in more detail later, these "tactical" goals are central to our theory of language: an utterance is not deemed to be comprehended until some direct consequence of it is seen as serving a goal imputed to the speaker. Furthermore, although the goals of the Components are active only within the conduct of a particular game, they are so constituted that their pursuit satisfies the goals described the Parameter Specifications, which were held by the participants prior to the evocation of the Dialogue-game.

In the case of the Helping-game, the goals in the "diagnostic" phase are that the HELPEE describe a sequence of related, unexceptional events leading up to a failure of his expectations. These model the state of the HELPER as he assimilates this initial part of the dialogue, both in that he knows how the HELPEE is attempting to describe his problem, and also that the HELPER knows when this phase is past, and the time has come (the "treatment" phase) for him to provide the help which has been implicitly requested.

Termination

The processes described above model the identification and pursuit of Dialogue-games. How, then, are they terminated? As we said previously, the Parameter Specifications represent those aspects of dialogues that are constant over that particular type of dialogue. The Dialogue-games model pushes this a step further in representing that the dialogue type continues ***only as long as*** the parameter Specifications continue to hold. Whenever any predicate in the Specification ceases to hold, then the model predicts the impending termination of this Dialogue-game.

For example, if the HELPEE no longer wants to perform the TASK (either by accomplishing it or by abandoning that goal), then the Helping Dialogue-game terminates, corresponding to the subsequent termination of the helping interaction. If the HELPER becomes unwilling to give help, or discovers that he is unable, then the Helping-game also terminates. Again, we have one simple rule that covers a diversity of cases--a rule for termination that captures the variety of ways that the dialogues we have studied end.

THE DIALOGUE MODELING SYSTEM

In this section we describe the major elements of the Dialogue Modeling system. In describing this system, we first show all the major parts, and their connectivity, in Figure 1. Next, these parts (two memories and six Processes) are each described separately. In an appendix there is an extensive, detailed trace of the system as it analyzes (via hand simulation) a naturally occurring dialogue fragment. Finally, we summarize our experience with the system to date.

(Note: for the observant reader who has noticed that there seems to be no "output" for this system, it should be pointed out that, according to the methodology we have employed (see below), the results of a completed analysis of a dialogue are to be found in a "post-mortem" examination of the sequence of memory states which held for the WS. Thus, in the course of running the analysis, there was no need to output any of the conclusions of the analysis.)

Long-Term Memory (LTM)

The Long-Term Memory is the system's model of one participant's knowledge of the external world. Here is held the grammatical case frames, the semantic structures for word-senses, the knowledge of the subject matter of the dialogue, the various ways in which dialogues are structured, the initial knowledge states of the participants, in short, everything we need to assimitate the dialogue, to the level of our aspirations.

LTM is a semantic memory, containing a set of nodes (also called concepts) and the relations that hold between them at the lowesi level. This information is stored in the form of triples:

<node-1 relation node-2>

We already have this machinery encoded and working--a full complement of read and write primitives for this representation. However, it has proven awkward for us to specify knowledge at this level, so we have implemented the machinery (named SIM) to translate n-ary predicates into these triples. Thus, for a predicate P having arguments A1, A2 and A3, when SIM is given the structure

P1: (Alpha P Beta Gamma)

[meaning that P1 is defined to be an instance of P (the predicate always goes in second position) with arguments Alpha for A1, Beta for A2 and Gamma for A3.] The resulting triples are created:

<P1 PRED P>	<P PRED-C P1>
<P1 A1 ALPHA>	<ALPHA A1-C P1>
<P1 A2 BETA>	<BETA A2-C P1>
<P1 A3 GAMMA>	<GAMMA A3-C P1>

Note that for each new triple, <a r c>, there is also created another triple which represents the "reverse-link", using the converse (-c) relation: <c r-c a>. Consequently, having the relations represented in both directions, the system is able to

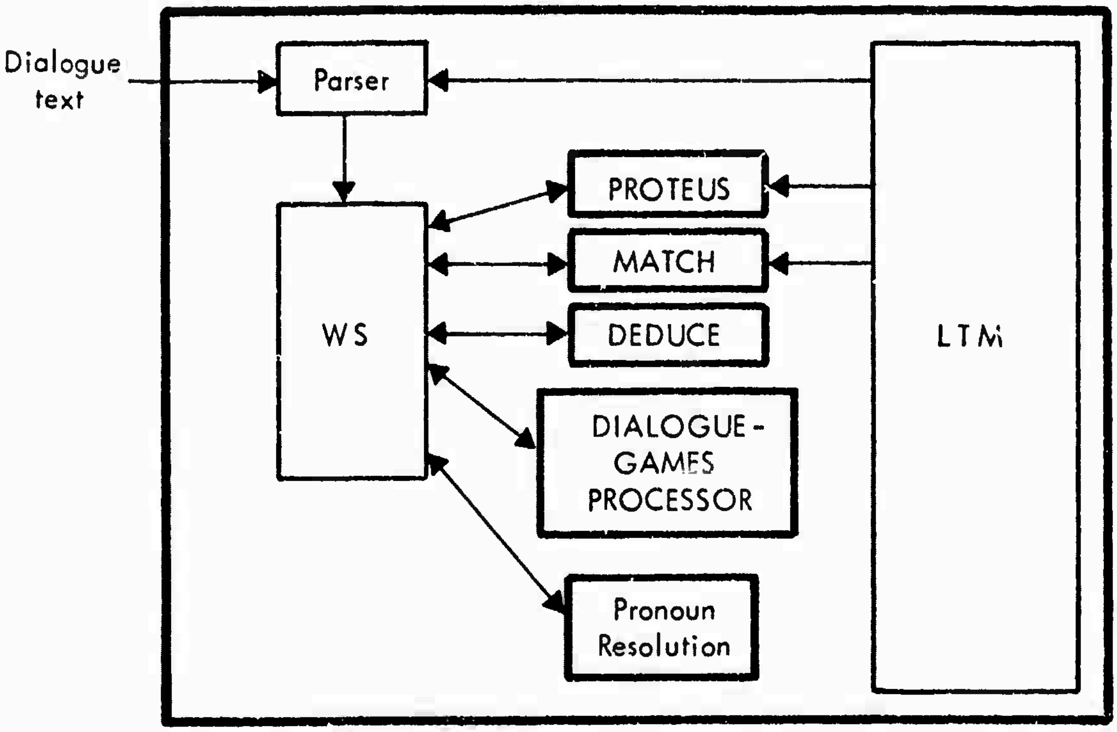

Figure 1. Dialogue analyzer

traverse the memory from any node to any other, to which it bears any relation, however indirect.

Let's examine a more concrete example; suppose we want to include in the LTM that:

Mary hit John with a rock.

The predicate "HIT" has two mandatory arguments (subject, object) and an optional one. The SIM representation of this assertion (which we shall name Q1) is

Q1:(MARY HIT JOHN ROCK)

which translates into the following triples:

<Q1 PRED HIT>	<HIT PRED-C Q1>
<Q1 SUBJ MARY>	<MARY SUBJ-C Q1>
<Q1 OBJ JOHN>	<JOHN OBJ-C Q1>
<Q1 INST ROCK>	<ROCK INST-C Q1>

For a node to be "in LTM" means that on a list named "LTM" is to be found a pointer to that particular node. With each node is stored the relation and second node (as an attribute/value pair) for each triple which involves that node.

Workspace (WS)

The Workspace is the system's model for that information which the participant is actively using. This memory corresponds roughly to a model of the participant's focus of attention.

Where the LTM is static during the operation of the model (we are not attempting to simulate learning), the WS is extremely volatile, with its items (activations) coming into and out of focus continuously. All incoming sensations (i.e., utterances) appear in the WS, as do all augmentations of the participant's knowledge and goal state.

The representation of information in the WS is the same as in LTM. As above, an activation is "in the WS" if a pointer to it appears on a list named "WS". The same access programs which work for concepts in LTM also manage activations in the WS.

In addition, each node in the WS is a token (copy) of some concept in LTM. Whenever some process determines that the time is ripe for the system's attention (WS) to include a token of a specific concept (C) from LTM, a new node (A) is created by copying C and this new node is added to the "WS" list. This is called creating an activation of C. A is referred to as an activation of C, and this relation is stored as

<A IAO C> and <C IAO-C A>

This representation provides the associative links between an object in attention, and the body of knowledge associated with it, but not yet brought into attention.

Parser

This module produces activations representing each successive utterance to be processed. These representations are generated from the surface string using a standard ATN Grammar similar to those developed by Woods (1970) and Norman, Rumelhart, & the LNR Research Group (1975). We use a case grammar representation, with each utterance specified as a main predicate with a set of Parameters. Because this module is a conventional parser whose implementation is well understood, we have so far produced hand parses of the input utterances, following an ATN grammar.

Proteus Process

This is a spreading activation mechanism, which modifies the activation of concepts specified as closely related in LTM whenever a given concept becomes active. This mechanism provides a way to integrate top-down and bottom-up processing within a uniform framework (Levin, 1976). The Dialogue Model System uses the Proteus Process to activate a concept, given that a number of closely related concepts (Components, features, instances, etc.) are active.

The Proteus Process operates on all current activations to modify their rate of activation. This is a number associated with each activation that generally represents the salience or relevance of the concept. There are two kinds of influence relations that can exist between concepts: excite or inhibit. If an excite relation exists, then the Proteus Process increases the rate of activation of that concept in proportion to the rate of the influencing concept. The higher the rate of an activation, the larger its influence on directly related concepts. If an inhibit relation is specified, then the Process decreases the activation rate of the neighboring concept.

Match

This Process identifies concepts in LTM that are congruent to existing activations. The Dialogue Modeling System contains a number of equivalence-like relations, which Match uses to identify a concept in LTM as representing the same thing as an activation of some seemingly different concept. Once this equivalent concept is found, it is activated. Depending on how this concept is defined in LTM, its activation may have effects on other processes (for example, if the concept is part of a rule, Deduce may be invoked).

Match can be viewed as an attempt to find an activation (A) in WS and a Concept (C) in LTM which correspond, according to some set of criteria. For purposes of this description, it is assumed (as a crudo ap ɔximation) that each Activation is paired with each Concept, and then for each such pair, a match is attempted. What follows is a description of how a Match is attempted for a single such pair.

The basic tactic is to attempt to find a form of equivalence relationship between A and C, without delving into their structure at all. Only if this fails are their respective substructures examined. In this second case, the same match which was attempted at the top level is tried between corresponding subparts of A and C. Match proceeds in five steps:

1. Is it already known that A is an activation of C? If so, the match terminates with a positive conclusion.

2. Is there any other activation (A') and/or concept (C') such that A' is known to be a view of A, C is known to be a kind of C', and A' is known (by step 1) to be an activation of C'? The relations (... is a view of ...) and (... is a kind of ...) represent stored relations between pairs of activations and concepts, respectively. One concept "is a kind of" another concept represents a superclass inclusion, true for all time and contexts. (Whatever else he might be, John is a kind of human being.) On the other hand, one activation may be "a view of" another only under certain circumstances--a conditional, or tactical relationship. Under different conditions, it is appropriate to view John as a Husband, Father, Child, Help-seeker, Advice-giver, etc.

3. There is a list of matched pairs of activations and concepts which represent correspondences found elsewhere, with which this match must be consistent. (N.B.: this Match, as we will see later, may be in service of another Match called on structures containing the current A and C.) If the pair [A,C] is a matched pair, then these two have been previously found to match, so we may here conclude the same thing and Match exits.

4. On the other hand, if there is either an X or a Y such that [A,X] (or [Y,C]) is a matched pair, then replace this match with an attempt to match C and X (or A and Y).

5. Finally, if the match has neither succeeded nor failed by this point, then Match is called recursively on all corresponding subparts of A and C, pairwise. That is, e.g., if A and C have only three subparts in common (say, SUBJ, OBJ and PRED) then Match((SUBJ of A),(SUBJ of C)), Match((OBJ of A),(OBJ of C)) and Match((PRED of A),(PRED of C)) are attempted. Only if all of these subordinate matches succeed is the top-level Match said to succeed.

Clearly, for structures of significant complexity, Match may eventually call itself recursively, to an arbitrary depth. However, since each subordinate call is on a strictly smaller unit, this process must converge.

Our experience has shown us that this type of mechanism plus a collection of rewrite rules enable us to eventually map a wide variety of input parsing structures to pre-stored, abstract knowledge structures, in a way that a significant aspect of their intended meaning has been assimilated in the process.

Deduce

This operates to carry out a rule when that rule has become active. Rules are of the form (Condition)->(Action), and Deduce senses the activity of a rule and applies the rule by activating the concept for the action. Whatever correspondences were evolved in the course of creating the activation of the condition (left) half of the rule are carried over into the activation of the action (right) half. The combination of Match and Deduce gives us all the capability of a production system.

The operation of Deduce is relatively simple It is called only when there is an activation of a rule in the WS. Deduce attempts to match the left half of this rule with some other activation in the WS. (This has typically already been done by Match, as we will see in detail, below.) Assuming this is accomplished, Deduce creates an activation of the right half of the rule, substituting in the activation for all subparts for which there has been found a correspondence in the left half.

Dialogue-games Process

Once a Dialogue-game has been activated as possibly the communication form being bid for a dialogue, the Dialogue-game Process operates on it to verify that the Parameters are properly specified, and then to establish the subgoals that are specified in LTM as the Components of the particular Dialogue-game. The Process then proceeds in four steps to:

1. establish/identify the Parameters of the game,
2. verify and/or assert the Specifications,

and

3. establish the Components as goals of the participants.

When the Process accesses each of the Parameters, they are found either to have activations in the WS or not. If they do, the correspondences between activation and Parameter are established in the WS. Any that has no activation is put on a list which is periodically checked in the hope that later activity by the Process will lead to the creation of appropriate activations.

For each of the Specifications, a check is made to determine if it already has an activation in WS. (In most cases, the activation of some of these Specifications will have led to the activity of the game itself.) The Specifications for which there are activations need no further attention.

Activations are created for all remaining Specifications, substituting for the Parameters as determined above. At this stage, the Dialogue-game Process calls Proteus to determine the stability of these new activations. Any new activation which contradicts existing activations will have its level of activity cut sharply by an iteration of Proteus. If this happens, the Dialogue-games Process concludes that some of the necessary preconditions for tho game do not hold (are in conflict with current understanding) and that this particular game should be abandoned. Otherwise, the new activations stand as new knowledge, depending on the hypothesis that the chosen game is appropriate.

On the assumption that this choice of Dialogue-game has been appropriate, the Process then creates activations of the game's Components, with appropriate substitutions. (By this time, any unresolved Parameters may well have activations, permitting their resolution.) This sets up all of the game-specific knowledge and goals for both participants.

Pronoun Processes

The Dialogue Model System contains a set of Pronoun Processes, including an I-Process, a You-Process, and an It-Process. Each of these is invoked whenever the associated surface word appears in an input utterance, and operates to identify some preexisting activation that can be seen as a view of the same object.

Each of these Processes search the current context, as represented by the current set of activations in the WS, using the features specified there to identify a set of possible co-referential expressions. When there is more than one possibility, the one with a higher rate of activation is selected.

RESEARCH METHODOLOGY

The model presented above was developed to exhibit specific pre-identified capabilities under a controlled evaluation method. This section describes the method, showing how it leads to objective comparison of models and evaluation of models and parts of models relative to human performance.

The method is based on case analysis rather than design of a system to perform a general function. Naturally occurring dialogues are collected before modeling starts. They are used as the model input--the text which the model must process. The model, after it is built, is thus subjected to the full complexity of some naturally occurring communication, but under circumstances in which it can easily be made capable of desired responses.

Modeling proceeds by a series of experiments with human dialogue, resulting in the creation of processes (represented as computer programs) which collectively are able to follow and make sense of dialogue transcripts. Since this approach could be pursued with varying scopes and degrees of detail, it is important to limit the range of phenomena. We exclude all of the long-term communication effects, from the multi-century development of language up to the personal. Long-term effects, including the gross developments of language acquisition, are outside our scope. At the detailed levels, articulatory or phonological or auditory phenomena are also outside the scope. Our smallest unit of analysis is the word or morpheme. Our goal is to understand the immediate effects of communication with words.

The effort addresses a limited range of described levels. The cycle of experiments works as follows:

1. In each single experiment, processes are developed which can cope with the transcript of a single dialogue. These processes are a kind of empirical hypotheses.

2. The results of multiple experiments are examined to identify those processes which recur, successfully coping with phenomena from several dialogues. These are the verified processes which we expect to successfully transfer into future communication systems. In multiple experiments a kind of refinement and verification of hypotheses occurs, with hypotheses stated as processes and debugged as computer programs.

SINGLE EXPERIMENT

Dialogue process modeling is a series of experiments which develop communicating processes. The single experiment consists of four parts:

1. Capture human dialogues.

2. Identify important phenomena in the dialogue.

3. Create processes that can follow the dialogue.

4. Evaluate the processes relative to the phenomena identified in item 2.

Each of these is explained in detail below. Figure 2 below illustrates the experiment.

Experiment Step 1

Step 1 is to capture the dialogue. A transcript of the dialogue, suitable for both computer-reading and human reading, is produced. The remainder of the experiment deals with this transcript, in particular with reconciling a human reading and a computer reading of it.

This step (and all prior steps) also includes some selection of transcript material for further analysis, since dialogue is abundant and easy to create, and there is always far more than we can analyze. This is one of several controls we have on the difficulty of our task. Another is the length of the exchange that we select.

To help follow the explanation, consider the following example, which is an actual operator-user dialogue, from the ISI TENEX computing facility, cleaned up for readability but essentially unaltered in content. This dialogue, between parties O (a computer system operator) and L (one of the system users), was mediated by the computer system. O and L each were typing at separate locations; RUNOFF is a commonly used text-formatting program. Spelling, punctuation and visual format have been edited for readability and privacy.

****

L:

How do I get RUNOFF to work, I keep executing it but it just grabs my input file and then says done but gives me no output? Go ahead.

O:

The output comes out on the line printer

L:

Throw it away but can I get it to go to a file? Go ahead.

O:

Confirm your commands with a comma and you'll be queried for files, etc. Go ahead.

L:

Thanx mucho

BREAK

****

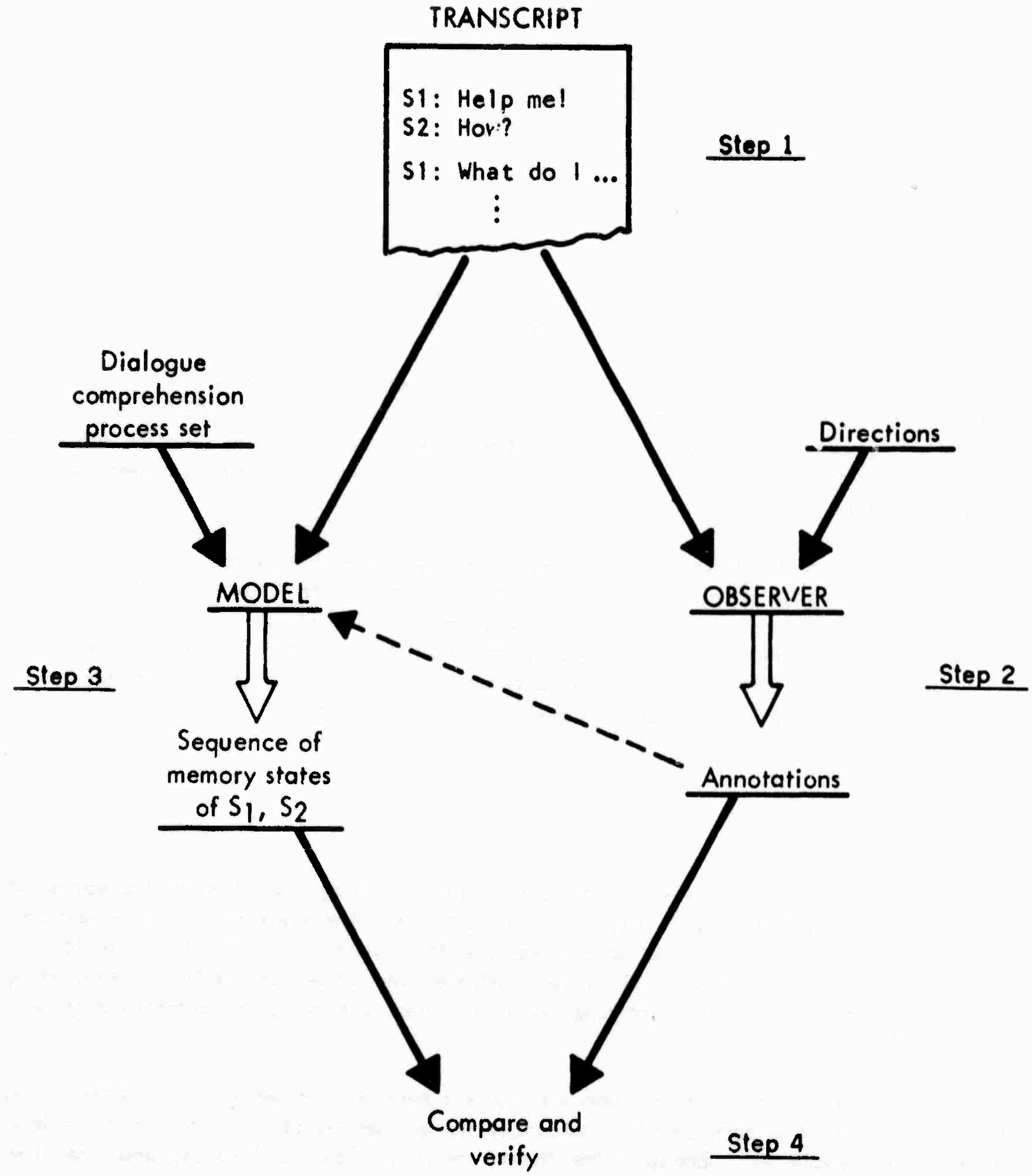

Figure 2. An experiment with a single dialogue

Experiment Step 2

Of course, a dialogue is generally a unique, one-of-a-kind event. It contains systematic features that make communication possible, together with a lot of unexplainable, idiosyncratic detail. Understanding the regularities that govern a dialogue requires describing them in common terms. The first stage of description in common terms involves deriving systematic data from dialogue. For this work, we use a person we call the Observer.

Step 2 in Figure 2 shows the Observer creating a commentary on the dialogue, based on a set of categories of observation described below.

The Observer is asked to assert only those phenomena for which he has high confidence that his interpretation would be widely agreed upon, say, by 95% of a group of people as competent as himself in communication. We do so in order to keep the attention of the modelers on the important phenomena. There is presently no profit in dealing with marginal cases.

> "Even though few reliable operational procedures have been developed, the theoretical (that is, grammatical) investigation of the knowledge of the native speaker can proceed perfectly well. The critical problem for grammatical theory today is not a paucity of evidence but rather the inadequacy of present theories of language to account for masses of evidence that are hardly open to serious question." (Chomsky 1965)

The Observation Process

The role of the Observer is not to tell why the dialogue went as it did, nor what methods the participants used to create their parts, since these are theoretical questions. Such questions belong to the accounting for the observations rather than to the observational process. Figure 3 below illustrates the Observer's activity, annotating transcripts of the dialogue according to instructions and his own understanding and judgement.

The Observer of communication activity is a kind of instrument. His judgments on the condition of the communication constitute the properties of the communication that need to be explained. Because the theory rests on "obvious" cases, and that the observation process be easy to understand and use without extensive training, the Observer is instructed not to annotate doubtful or unclear cases.

The Observer is a kind of reducer of the data, an inscrutable perceiver and filter, whose presence is acceptable and necessary because his comments somehow represent his whole community of communicators, and therefore represent underlying regularities which arise from the conventional communication methods of that community. Thus in this methodology the observer must be a person; he cannot be replaced by a computer program.

The Observer is able to answer sufficiently specific questions about the dialogue with high reliability and repeatability, and is normally able to state correctly the

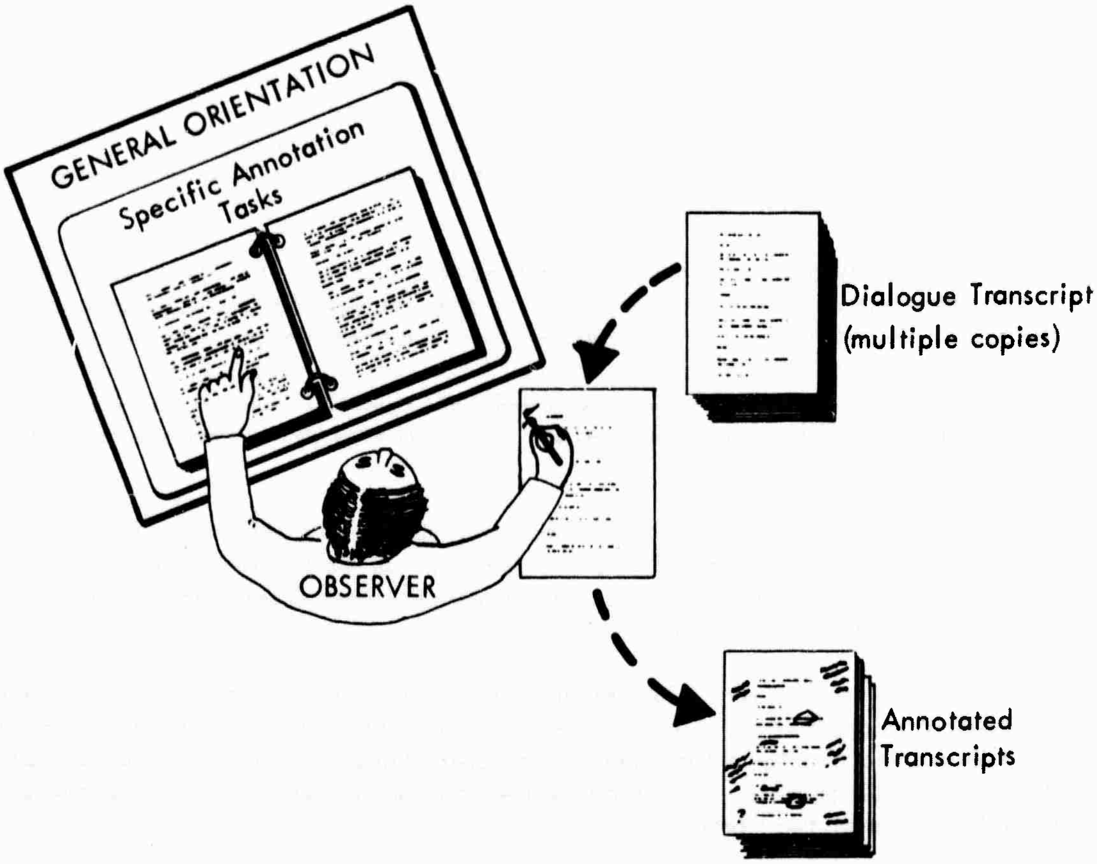
GENERAL ORIENTATION
Specific Annotation Tasks
Dialogue Transcript (multiple copies)
OBSERVER
Annotated Transcripts

Figure 3. The Observer's task

communication effects of parts of the dialogue.* Of course, the reliability and repeatability of Observers must be verified experimentally (cf. Mann, et. al. (1976)). If they were not reliable, then there would be no basis for claiming that a model conforming to them was comprehending language in conformity to people's comprehension.

In order to have the observations in an experiment reflect a single coherent point of view, each dialogue is modeled relative to the comments of a single observer rather than with pooled, possibly inconsistent, observations.

Part of the work of the project is to develop this framework of observation into an easily used tool. The Observer is trained to seek several prescribed categories of phenomena, which are described below. The function of each of these categories of observation is to influence the attention of the model-builders toward some aspect of ordinary, successful human communication. By attending to a suitable number of cases, the model builders are to derive processes whose structure reflects the regularilies of at least those cases, and hopefully much more. Since each of these models must account for multiple categories of observations, interactions and mutual dependencies among people's methods are reflected in the processes.

Some parts of the dialogues are not addressed directly by any observation, but processing them is nevertheless necessary as a prerequisite to successful processing of the rest. Processes may be involved that only contribute indirectly to accounting for observations. (For the categories described below, parsing processes are such.) These processes have been partially developed *and their execution contexts and consequences identified. Therefore they can be verified just as if they were directly accounting for varieties of observation.*

So the particular observation categories chosen are influential, but ultimately arbitrary. They are not primitives, and they do not necessarily combine to give a complete picture of human communication. Experience over several more dialogues may suggest that one or all be replaced.

There is actually very little risk that the present formulation fails to get at the key communication phenomena. If there were some basic sort of process whose effects were not represented directly in the scope of the observations, it would be modeled anyway if it had major consequences in the scope of the observations, which is just what one would expect of any such basic process.

----

* The Observer is doing a kind of encoding of the interaction. There is a line of development of group interaction coding methods in social psychology, including for example Bales (1951). These developments are of some help, but our demands (for example, relative to content) are quite different.

Categories of Observation

The Observer is trained to identify seven more-or-less independent kinds of phenomena:*

1. Repeated Reference
2. Requests
3. Expression of Comprehension
4. Topic Structure
5. Correction Actions
6. Similar Expressions

Repeated Reference deals with what is mentioned or referred to. It includes the actions of nouns, pronouns and verbs in English. We deal with two kinds: ordinary repeated reference and text reference. In order to get reliable observations, we ask the Observer to identify multiple mentions of the same thing, rather than requiring him to spell out just what is referred to.

For example, the Observer noted that "RUNOFF" and the first 2 occurrences of "it" refer to the same thing in the example dialogue above.

The Requests category deals with all kinds of real and apparent behavior-seeking in dialogue. The Observer makes detailed annotations about 5 subcategories:

1. Questions -- request immediate, verbal response.
2. Orders -- request immediate, nonverbal response.
3. Directives -- request certain behavior in the future.
4. Rhetoricals -- look like Requests but are not.
5. Prohibitives -- request *to not do* something.

The Observer annotates how the Request occurs, and also how in most cases how the partner responds. In the example, "can I get it to go to a file?" is one of the Questions the Observer found, and he noted that eventually it was answered.

Expression of Comprehension deals with the direct and indirect evidences given by one partner to the other that he has comprehended something which has gone before. In the exampte, O's way of speaking of "output" indicated that he comprehended what L said.

Topic Structure deals with the way topics of conversation are begun, acknowledged, carried on, suspended and dropped. The short example above has only one topic, which persists over the whole dialogue.

Correction actions are parts of the dialogue in which one party cancels or changes some previous part of the dialogue that has already been understood by each party (often in two different ways.) There are no correction actions in the example above.

----

* All of the work of the Observer is described in detail, with definitions, instructions and examples, in a recent report (Mann, et. al., (1975)).

The idea of Similar Expressions is related to the idea of paraphrase. Tho Observer scores alternative ways of saying what was said, judging whether they would be suitable substitutes under certain circumstances. The intent here is to force the model to represent underlying functional equivalences rather than making it responsive only to surface similarities between expressions in communication. The two kinds of Similar Expressions arise because under one condition, the Observer judges whether the expressions are suitable alternatives in any imaginable circumstance, and in the other condition he judges whether the expressions are suitable alternatives in the context of a specific dialogue.

In the example above, O might as well have said "RUNOFF sends its output to the printer" in the second turn. In some contexts, "Get it away from you by throwing it" might function as well as "Throw it away" , but in this context it would not. These are representative findings from the observation of Similar Expressions.

In every category, the Observer is instructed to assert only on cases that he feels are clear and easy to decide. We avoid putting much attention on marginal cases.

Experiment Step 3

The next major step (Step 3 in Figure 2) is to create a process model (a new computer program for this specific dialogue) that copes with the dialogue transcript. Informally, this means that the model must *maintain a simulated knowledge state for each participant* that is adequate for supporting continuing interpretation of the dialogue. The adequacy of this continuing interpretation is assessed by the Fidelity evaluation and Recurrence evaluation methods described below.

The program must follow the dialogue in enough detail so that references of pronouns and noun phrases are identified, requests and how they are resolved are identified, corrections are accurately identified, topic flow is correctly assimilated and so forth.

The model is *not* attempting to simulate the Observer. It is simulating *the receptive acts of the participants* in the dialogue. The model takes as input the transcript of the dialogue, with the text produced by each party identified. It has a portion of its memory allocated as a simulated memory for each of the participants. The model's task is simply to keep these memories up to date.

The simulated memories of the participants are structured to represent different kinds of knowledge and knowledge status. Current awareness is distinguished from other available knowledge. Knowledge which a participant believes is not known to the other participant is distinguished from knowledge which he regards as shared. Linguistic knowledge, facts about the world, and knowledge about the environment of the dialogue are distinguishable. Each simulated participant has in memory a model of the other party and of himself. Each party's goals and abilities are represented. Of course, the depth of representation depends in each case on the demands of the dialogue.

We have a strong advantage here over most programming activity in that all of the performative demands on the program to be produced are explicit and available when the program is written. The input (the dialogue transcript) and the standard of evaluation (the Observer's commentary) are open to inspection. (Hindsight is clearer than foresight.)

People's ability to anticipate the contingencies of program input is rather limited. This is one reason that debugging is the major cost component of commercial programming. Having the input at hand is much more efficient.

Also, we make use of methods that are known to be inadequate for the general case. We explore their function for specific successful cases, and either generalize after gaining experience or find a set of methods that jointly cover the requirement.

Another advantage is that each dialogue is a fresh problem. There is not a backlog of design commitments, habits, and development cost which must be accommodated. From dialogue to dialogue, we keep what works and drop the rest. The amount of work carried forward from one dialogue to the next depends on their underlying similarity. This is another controllable feature, since we select dialogues and dialogue sources as we go along. From the point of view of scientific method, we are being driven by the data rather than by anticipations of what the data might be, or by anticipations in the form of system specifications. We find that this practice keeps the focus away from peripheral and artificial problems.

Experiment Step 4

Step 4 of the experiment is called Fidelity Evaluation. It is a comparative evaluation of the process model relative to the Observer's commentary. The purposes of the evaluation are:

1. To identify program states and actions which are in agreement with the Observer's comments.

2. To identify Observer's comments for which the corresponding process states and actions are either absent or disagree with the observations.

3. To identify the directly contributing processes whose states and actions are in agreement with the Observer's comments.

The comparison is between the observations of a dialogue and the trace of the model for that dialogue. (The trace is a sequential record of the actions of the model during the interpretation of the dialogue.) For each observation which the Observer asserted, the trace is examined to see whether there is a corresponding action or partial state of the model. If so, then the observation is counted as having been successfully modeled. The time (in the trace) at which the observation was fulfilled is identified.

For each successfully modeled observation, the trace is examined to identify the directly contributing processes. These are usually those which are active at the time of fulfillment of the observation, together with the processes which produced the conditions, knowledge or data which controlled the branching of the model's control flow immediately preceding the time of fulfillment.

Multiple-Experiment Comparisons

The Need for Comparison

A successful accounting for a single dialogue yields a set of processes of unknown value. The value of the processes, both as a scientific account and as components of future systems, depends on their generality, i.e., their effectiveness on different communications, different environments and communication goals. The high-value processes are those which have high coverage of the scope of human communication (illustrated in Figure 4 below). There is a spectrum of values from the totally ad-hoc process which can cope with only one dialogue, to the totally general process which copes with every case in its domain. The next step, Multi-experiment comparison, is intended to identify processes which, in our sample of experiments, have been found repeatedly effective.

We identify our high value processes as those which are reasonable on a variety of dialogues. A diversity of dialogue sources, media and goals in our experiments is necessary in order to get a good measure of value for our processes. This diversity also serves the system designer using the processes, since it gives him the evidence that they are reliable and transferable.

Recurrence Evaluation Method

The Recurrence Evaluation method is quite simple. At the end of each experiment, we identify those processes that have been:

1. used unaltered from previous experiments, or. could be retrofitted into previous experiment.

2. identified as contributing to successful accounts of the Observer's comments.

These processes are considered to have demonstrated value.

How Recurrence Evaluation Drives Algorithm Development

The comparison in Recurrence Evaluation has a pervasive effect on the whole modeling process. Any processes that are dependent on features of a particular dialogue, or that work in a known-unreliable way, will not show demonstrated value at this step. ***The goal in modeling is to maximize the product of this comparative step.*** This leads the modelers to include general features in processes whenever it is sufficiently convenient to do so, to seek the unifying principles, and to represent them in models. It avoids the problem of creating many independent, incommensurate and finally useless single-case models.

Deriving Valuable Results

There are several approaches to the products of this methodology, depending on what is regarded as valuable. The activity can be supported from various viewpoints, of which we consider two briefly:

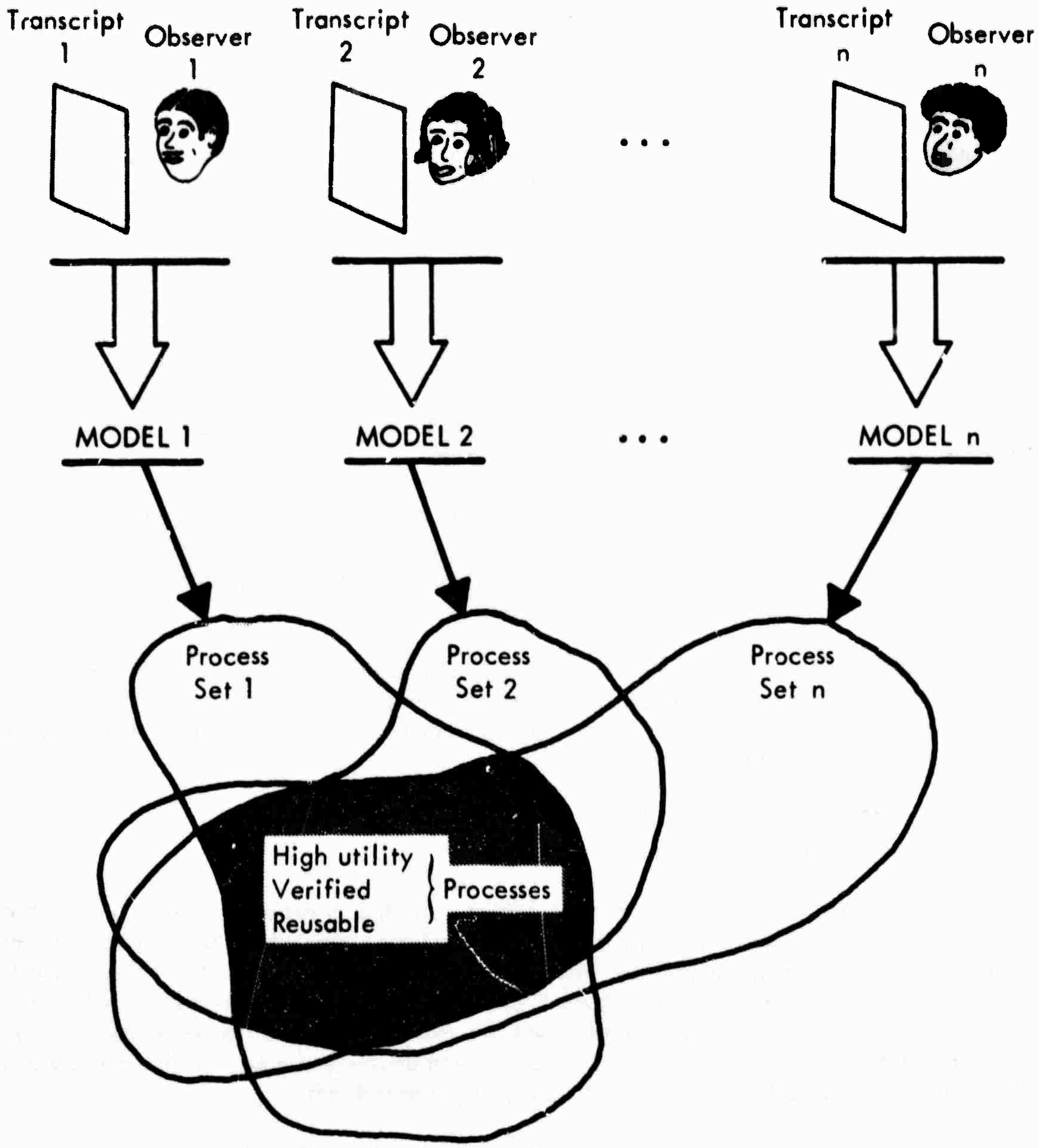

Figure 4. Evaluating multiple experiments

1. Scientific Knowledge
2. Application in Computer Systems

For both of these, the primary results are processes rather than whole systems, since the individual process is a decisively more convenient unit of investigation or application than the enclosing system in which it occurs.

Scientific Values

We have previously discussed several branches of science which express their theories in terms of discrete symbolic processes like those we develop. In others (e.g., the medical theory of communication disorders and brain damage) such theories are possible but seldom found. In any of these, the processes we develop should be considered as theories of phenomena of that branch.

The evidence for the effectiveness and generality of a process is explicit and easily examined -- it is the dialogues and observations which the process has contributed to accounting for, and the model traces which exhibit the manner of contribution. The empirical approach of this method makes its results directly relevant.

Beyond this, we anticipate that this approach will eventually have an integrating effort, providing common representations across discipline boundaries and coordinating knowledge developed in the separate disciplines. For this purpose, the general framework and its processes are both of interest.

Application in Computer Systems

One reason for designing the methodology so that it produces processes is that processes are primary components of computer systems. We anticipate relatively direct transfer of convenient processes into working system environments. The earliest applications are planned as modifications of existing systems. (For example, a process for correcting one's previous statements might be moved from a dialogue model into an existing text editing program so that the editor could understand a method that people use for correcting themselves, applying it to previously entered text.)

This "Method of Embedding" provides a very direct transfer of the technical results into applications, which was one of the goals described above. Algorithms are selected informally for embedding.

Of course, the process can be designed into new systems as well. Again, the transfer is very direct, especially relative to research that produces only factors for the designers' consideration.

DEFICIENCIES IN CURRENT MAN-MACHINE COMMUNICATION

With the understanding we now have of the super-sentential aspects of human communication, it is easy to see why man-machine communication appears so alien, highly restrictive, uncomprehending and awkward. This is because ***major regulation and interpretation structures are missing.***

In Table 1, we compare human dialogue and typical man-machine communication with respect to some of these features. The table designates a "sender" and a "receiver" which should be identified with the person and the computer respectively in the man-machine communication case. (This puts the man-machine interface in the most favorable light, since typically the man-to-machine communication is schematized by parametric command languages, and the machine-to-man communication is not schematized at all.)

TABLE 1
A COMPARISON OF MAN-MAN AND MAN-MACHINE COMMUNICATION

ASPECTS OF NATURAL COMMUNICATION ADDRESSED BY DIALOGUE-GAME THEORY	HUMAN DIALOGUE	MAN-MACHINE
Sender's goals known to recipient	YES	NO
Participants can declare goals and commands	YES	NO
Goals persist over several messages	YES	NO
Goals identified with each message	YES	NO
Communication plans used	YES	LITTLE
Implicit communication takes place	YES	LITTLE

Conventional man-machine communication frequently gives the user a sense that the computer is operating "out of context", since he must continually respecify what is relevant to the ongoing dialogue. In human communication it is the shared awareness of each other's goal structures which permits them to retain and focus on what is relevant. Man-machine communication seems aimless and undirected because there is no analogous body of knowledge being used to facilitate and interpret the communication.

SUMMARY AND CONCLUSIONS

This report has described a research program aimed at modeling human communication ability. In this final section, we summarize the highlights of this report.

First, we delimited the scope of our interests for this research. The behavior the models deal with is that of two humans engaged in a short dialogue, conducted over a restricted media so as to exclude visual and intonational communication. Further, these dialogues are selected to represent a "closed" discussion, i.e., one in which topics discussed are only those introduced within the behavior being examined, and these topics are pursued to whatever resolution they ever received. Within this restricted domain, the behavior we were specifically addressing was the receptive aspects only.

Next, we introduced a new theory addressing certain aspects of the human use of natural language. In particular, these persons were seen as instances of information processing systems engaged in problem-solving behavior, in pursuit of their own collection of goals. Each has an image of the other which includes estimates of the other's knowledge and goal states, abilities, inclinations toward actions, and probable effects on the other of his hearing certain utterances.

Our investigations of the dialogue data led us to the discovery and representation of recurrent goal/knowledge structures which seem to be shared by the participants in these dialogues, and which appear to characterize significant (and previously unaccounted for) phenomena of actual language usage. Specifically, these structure, called Dialogue-games, enable the model to account for a useful amount of the clearly present but implicit communication.

A system was designed which was to be the basis of the model building effort. This system contains a long-term-memory (LTM) which holds all the information needed at any point in the dialogue, and a workspace (WS) which constitutes the system's partial state of knowledge, goals, and focus at any point in time. There are a collection of processes to modify and update the workspace. Since the intent is to model receptive acts only, the model does not generate any text, or even any utterance goals. Instead, the "output" of the model, for our purposes, is simply a trace of the activity in its workspace, as it assimilates the dialogue. It is in this trace that the aspects of comprehension are to be found which correspond to the Observer's extractions, and which then constitute the evidence of the adequacy of this model for those particular dimensions of comprehension.

We described the methodology developed to explore this theory. Observers are used to extract features from naturally-occurring dialogues, and these features are used as target behavior, both to guide the development of the models, and as criteria against which to judge the finished product. A new model is to be built for each dialogue studied; the desired "exportable" results are those subprocesses (each dealing with some small part of the comprehension) which are found to be reusable in a significant number of these independent models.

Finally, as a result of this research, we were able to pinpoint certain aspects of natural language which are crucial to its use as a tool for communication, but which are completely absent in existing man-machine interfaces.

An extensive simulation of the system, at its current state of design, was conducted using a single dialogue as data. The results of this simulation (detailed in the appendix) are an exhibition of the sufficiency of the theory, and the design of its implementation, to the range of phenomena we were addressing. In particular, the simulation indicated that the model was capable of recognizing when a Dialogue-game was being bid, and what the consequences of that bid were, and what the initial and subsequent knowledge and goal states of the two participants were.

REFERENCES

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

Bartlett, F. C. *Remembering.* Cambridge, England: Cambridge University Press, 1932.

Bales, R. F. *Interactive process analysis.* Cambridge, MA: Addison-Wesley, 1952.

Bransford, J. D., & Johnson, M. K. Considerations of some problems of comprehension. In W. G. Chase (Ed.), *Visual information processing.* New York: Academic Press, 1973.

Charniak, E. Organization and inference in a frame-like system of common sense knowledge. In R. Schank & B. L. Nash-Webber (Eds.), *Theoretical issues in natural language processing.* Cambridge, MA: Bolt, Beranek and Newman, Inc., 1975.

Chomsky. N. *Aspects of the theory of syntax.* Cambridge, MA: The M.I.T. Press, 1965.

Clark, H. H., & Lucy, P. Understanding what is meant from what is said: A study in conversationally conveyed requests. *Journal of Verbal Learning and Verbal Behavior,* 1975, *14,* 56-72.

Erman, L. D, Fennell, R. D., Lesser, V. R., & Reddy, D. R. System organizations for speech understanding. *Proceedings of the Third International Joint Conference on Artificial Intelligence.* Palo Alto, CA: Stanford University, 1973.

Gordon, D., & Lakoff, G. Conversational postulates. *Papers from the Seventh Regional Meeting,* Chicago Linguistic Society, 1971.

Grice, H. P. Logic and conversation. In P. Cole & J. L. Morgan (Eds.), *Syntax and semantics.* New York: Academic Press, 1975.

Labov, W., & Fanshel, D. *Therapeutic discourse: Psychotherapy as conversation.* Draft copy, 1974.

Levin, J. A. Proteus: An activation framework for cognitive process models. Unpublished doctoral dissertation, San Diego, CA: Univ. of Calif, San Diego, 1976.

Levin, J. A., & Moore, J. A. Dialogue-games: A process model of natural language interaction. *Proceedings of the AISB Summer Conference.* Edinburgh, Scotland, July 1976.

Mann, W. C. Why things are so bad for the computer-naive user (ISI/RR-75-32). Marina del Rey, CA: Information Sciences Institute, 1975.

Mann, W. C., Carlisle, J. H., Moore, J. A., & Levin, J. A. An assessment of reliability of dialogue annotation instructions (ISI/RR-77-54). Marina del Rey, CA: Information Sciences Institute, 1977.

Mann, W C., Moore, J. A., Levin, J. A., & Carlisle, J. H. Observation methods for human dialogue. (ISI/RR-75-33), Marina del Rey, CA: Information Sciences Institute, 1975.

Minsky, M. A framework for representing knowledge. In P. H. Winston (Ed.), *The psychology of computer vision*. New York: McGraw-Hill, 1975.

Newell, A. Production systems: Models of control structures. In W. G. Chase (Ed.), *Visual information processing*. New York: Academic Press, 1973.

Newell, A., & Simon, H. A. *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall, 1972.

Rieger, C. The commonsense algorithm as a basis for computer models of human memory, inference, belief and contextual language comprehension. In R. Schank & B. L. Nash-Webber (Eds.), *Theoretical issues in natural language processing*. Cambridge, MA: Bolt, Beranek and Newman, Inc., 1975.

Rumelhart, D. E. Notes on a schema for stories. In D. G. Bobrow & A. Collins (Eds.), *Representation and understanding: Studies in cognitive science*. New York: Academic Press, 1975.

Sacks, H., Schegloff, E. A., & Jefferson, G. A simplest systematics for the organization of turn-taking for conversation. *Language*, 1974, *50*, 696-735.

Schank, R. C., & Abelson, R. P. Scripts, plans and knowledge. Paper presented at the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, USSR, August 1975.

Searle, J. R. *Speech acts: An essay in the philosophy of language*. Cambridge, England: Cambridge University Press, 1969.

Searle, J. R. Indirect speech acts. In P. Cole & J. L. Morgan (Eds.), *Syntax and semantics*. New York: Academic Press, 1975.

Shatz, M. How young children respond to language: Procedures for answering. *Papers and Reports on Child Language Development*, Palo Alto, CA: Stanford University, 1975.

Thorndyke, P. W. Cognitive structures in comprehension and memory of narrative discourse. *Cognitive Psychology*, 1977, *9*, 77-110.

Wittgenstein, L. *Philosophical investigations* (3rd ed.). New York: Macmillan, 1958.

Woods, W. A. Transition network grammars for natural language analysis. *Communications of the ACM*, 1970, *13*, 591-606.

APPENDIX -- SIMULATION OF THE DIALOGUE MODELING SYSTEM

In this appendix we describe an extensive simulation of the current state of the Dialogue Modeling System. We make use of a particular version of the Helping-game and also explore another structure, an Execution Scene, which describes the customary events surrounding the successful execution of a particular program (Runoff).

We start by describing this more detailed version of the Helping-game, introducing names for the various aspects, to be used later. Next we show a short, naturally occurring dialogue between a computer operator and a user. Then we describe the operation of the Dialogue Modeling System as it assimilates this dialogue, up to the point at which it concludes that the Helping-game is an appropriate structure through which to understand the subsequent utterances.

Once this hypothesis for the form of the dialogue has been chosen, we continue the simulation to examine how the system decides that a particular Execution Scene is appropriate for assimilating the content of the dialogue. Next, we see how this choice of scenes enhances the set of goals imputed to the speaker, thus facilitating the comprehension of what he is saying. Finally, we summarize our experience with the Dialogue Modeling system so far.

A Detailed Structure for the Helping-game

What follows is the substance of the communication structure we have named the Helping-game. In the interests of clarity of presentation, the formal structures of the definition have been expressed in prose. However, there is a one-to-one correspondence between the elements of the following description and those in the actual Helping-game used in the simulation.

HELPING-GAME

Parameters:
The parameters are two roles (HELPER and HELPEE) and a topic (TASK/HG).

Parameter specifications:
The HELPER and HELPEE are each a kind of person.

H1 = A goal of the HELPEE is that he perform TASK/HG.
H2 = It is not true that HELPEE is able to perform this TASK/HG.
H5 = The HELPEE wants to be able to perform the TASK/HG.
(being able to perform the task is a subgoal of performing the task)
H6 = The HELPER is able to enable the HELPEE to perform the TASK/HG.
H8 = The HELPER is willing (= is able to want to ...) to enable the

HELPEE to perform the TASK/HG.

H10 = The HELPEE is permitted to perform the TASK/HG.

H11 = The HELPEE wants the HELPER to enable him to perform the TASK/HG.
(being enabled to perform the task is a subgoal of
performing the task)

Game components:

HGX1 = The HELPEE knows of a particular execution scene, XS/HE.
[note: an execution scene is a flowchart-like description
of the use of a particular process; more details below]

HGX2 = The HELPEE knows that his perceiving the terminal state of XS/HE
would satisfy his wanting to perform TASK/HG.

HGX2C= (Thus) The HELPEE wants to perceive XS/HE in this terminal
state.
(this perception is a subgoal of performing the TASK/HG)

ACTION/GOOD = an ACTION of XS/HE which was realized in the past.

HGX3 = The HELPEE knows he has perceived this ACTION/GOOD.

HGX4 = The HELPEE knows he had expected to perceive it.

HGX5 = The HELPEE knows he wants to perceive this ACTION/GOOD.
(perceiving the ACTION/GOOD is a subgoal of perceiving the
[desired] terminal state of the XS/HE)

ACTION/BAD = an ACTION of XS/HE which was not realized in the past.

HGX6 = The HELPEE knows that he did not perceive ACTION/BAD.

HGX7 = The HELPEE knows that he had expected to perceive it.

HGX8 = The HELPEE wants to perceive ACTION/BAD.
(perceiving the ACTION/BAD is a subgoal to perceiving the
terminal state of XS/HE.)

HGX9 = The HELPEE wants to describe what happened which was both
expected and wanted, the ACTION[s]/GOOD.
(describing these ACTION[s]/GOOD is a subgoal of having
the HELPER enable the HELPEE to perform the TASK/HG.)

HGX10= The HELPEE wants to describe what did not happen that he
expected, and wanted, the ACTION[s]/BAD.
(describing these ACTION[s]/BAD is a subgoal of having
the HELPER enable the HELPEE to perform the TASK/HG.)

The Dialogue to be Modeled

What follows is a transcript of a naturally occurring dialogue between a computer operator (identified as "O") and a user ("L") who has "linked" to the operator, in an attempt to solve a problem.

There has been virtually no "cleanup" of this transcript, except to remove extraneous typing that had appeared on the operator's console listing as a result of the operating system printing routine status messages. The choice of words, and even spelling, are exactly as typed by the participants. (We have segmented the text by interposing carriage-returns as we deemed appropriate.)

Dialogue OC117

LINK FROM [L], TTY 42

L: How do I get runoff to work,
I keep xeqtn it
but it just grabs my input file
and then says done
but gives me no output?
GA

O: The output comes out on the line printer

L: Throw it away
but can I get it to go to a file?
GA

O: Confirm your commands with a comma
and you'll be queried for files, etc.
GA

L: Thanx mucho
BREAK

The subsequent simulation is of the system processing the first five segments, the entire first utterance. Each utterance is ingested one at a time, by the Parser, and the analysis proceeds until a quiescent state is reached (much more detail, below) whereupon the next segment is parsed and input for processing.

The Identification of the Helping Dialogue-game

How does the model know to evoke the Helping Dialogue-game? To exhibit answers to this and subsequent questions, we lead the reader through a simulation of the model as it processes the beginning of dialogue OC117. We indulge in the same use of prose for formalism as above, again with the same assurances of correspondences with the actual simulation.

The simulation proceeds in cycles; in each cycle, we exhibit the operation of a single processor, performing one iteration of its function. We do not address here the issues of how the system would select which processor to call next. In fact, our design calls for these processors to be maximally autonomous and parallel in their operation, operating whenever circumstances are ripe for their function and dormant otherwise.

The format of this simulation is as follows: The cycle number is first, in the form: <segment number>-<cycle number in this segment>. Next is the name of the processor operating in this cycle. After that is a description of the nature of the processing done during that cycle. Finally, there is a list of the results for this cycle, that is. all the important changes in WS.

Initially, the description is at a very detailed level. But after a while, the operations become extremely repetitive so the description becomes less detailed, focusing only on the unique aspects of the current operation. In this example, each processor is called at least once in the processing of each segment; Match, Deduce and Proteus bear the major burden, having several invocations each per segment.

Cycle 1-1 -- Parse.

The parser reads one utterance/segment of input and translates it into the formalism for activations in the workspace. No claim is made that this translation retains all the content of the original text, only that it is adequately faithful to the level of detail we are simulating.

Results: Case/9 (= (O perceives that L asks (how do I get Runoff working?))) is activated.

Cycle 1-2 -- I-processor

Certain words (e.g. pronouns, determiners) are taken to be signals that a reference is being made to concepts introduced elsewhere. The presence of a concept in the workspace corresponding to one of these words leads to the calling of the process-specialist which attempts to resolve the implied reference. Thus, the presence of "I" in the text leads to the calling of the I-process, whose sole function is to determine the referent of the "I" and modify the stored concept to reflect this. This process judges that if L is asking a question which contains "I" as its subject, then this constitutes adequate evidence to hypothesize that "I" is being used to refer to L.

Results: O perceives that L asks (how does L get Runoff working?)

Cycle 1-3 -- Match

Match is always on the lookout for pairs of nodes, one in the WS and the other in the LTM, such that the activation (node in WS) matches the concept (node in LTM). This is taken to be evidence that the activation is also to be taken as an activation of the matched concept. It should be understood that we are examining only some of the successful matches which occurred.

Starting in this cycle, we see a pattern which recurs regularly, and which accounts for a significant piece of the action, as the model assimilates the dialogue. Match determines that a particular activation matches the left half (condition side, if part, etc.) of a production-like rule stored in LTM. This successful match leads to the identification of the correspondences between the aspects of tho activation and those of the left half of the rule, as well as creating an activation of the rule itself. The activation of a rule leads to calling the Deduce processor in the next cycle, which applies the activated rule to the node in the WS responsible for the rule's activation. This application of a rule (which also results in the removal of the rule's activation from the WS) creates a new activation structure in the WS.

In other words, the introduction of a piece of knowledge suggests that a certain transformation (e.g., "Whenever you know X, you can conclude Y.") is appropriate. This transformation is applied to the stimulus knowledge to generate a conclusion: a new piece of knowledge.

In this particular case, the above result structure is found to match the left half of

Rule0 = If O perceives a proposition,
then O knows that proposition.

with the correspondences
Case/1 (= (L asks (How do I get Runoff working?))) is activated.
corresponds to the proposition.

(This rule represents the approximation that what is perceived is accepted at face value.)

Since Case/9 is now seen to be an activation of the Left-half of Rule0, an activation for the rule itself is created in the WS.

Results: Case/9 is an activation of Left half of Rule0.
Case/1 corresponds to the proposition in Rule0.
An activation of Rule0 is entered into WS.

Cycle 1-4 -- Deduce

Since there is an activation of a rule in WS, Deduce is called in an attempt to apply the rule. The Match has guaranteed that the necessary correspondences exists between the left half of the rule and the node which is its activation. To apply the rule, Deduce creates an activation of the right half, with the corresponding sub-parts substituted.

Results: RO-1 = O knows Case/1
Activation of Rule0 deleted from WS.

Cycle 1-5 -- Match

Match finds that RO-1 matches the left half of:

Rule1 = If O knows (L asks about a proposition),
then O knows (L does not know about that proposition).

Results: RO-1 is an activation of the left half of Rule1.
Case/1 corresponds to (L asks about a proposition)
Case/2 = (How does L get Runoff working) corresponds to the
proposition.
An activation of Rule1 is created in the WS.

Cycle 1-6 -- Deduce

Deduce applies Rule1 to R0-1, substituting according to the discovered correspondences.

Results: R1-1 (= O knows (L does not know Case/2), is activated.)
Activation of Rule 1 deleted from WS.

Cycle 1-7 -- Match

Match R1-1 with left half of

Rule3 = If O knows that a person does not know how to perform a task,
then O knows that that person is not able to perform the task.

Results: R1-1 is an activation of the left half of Rule3.
L corresponds to the person mentioned.
Get corresponds to Perform.
The state of Runoff working corresponds to the task.
An activation of Rule3 is created in the WS.

Cycle 1-8 -- Deduce

Deduce applies Rule3 to R1-1.

Results: R3-1 (= O knows that R3-11 = (L is not able to perform (getting Runoff working)) is activated).
Activation of Rule 3 deleted from WS.

Cycle 1-9 -- Match

Match R3-11 with H2 = Helpee is not able to perform the task.

Results: R3-11 is an activation of H2.
(getting Runoff working) corresponds to the task.
L corresponds to the Helpee

Cycle 1-10 -- Match

Match R0-1 with left 1/2 of:

Rule2 = if O knows (L asks about a proposition),
then O knows (L wants to know about that proposition).

Results: R0-1 is an activation of the left half of Rule2.
Case/1 corresponds to (L asks ...), in Rule 2.
Case/2 corresponds to the proposition.

An activation of Rule 2 is created in the WS.

Cycle 1-11 -- Deduce

Deduce applies Rule2 to R0-1.

Results: R2-1 (= O knows (L wants to know about Case/2) Is activated).
Activation of Rule 2 deleted from WS.

Cycle 1-12 -- Match

Match R2-1 with left half of

```
Rule4 = If O knows (a person wants to know how
            to perform a task),
           then O knows (that person wants to perform that task).
```

Results: R2-1 is an activation if the left half of Rule4.
L corresponds to the person.
(getting Runoff to work) corresponds to the task.
An activation of Rule 4 is created in the WS.

Cycle 1-13 -- Deduce

Deduce applies Rule4 to R2-1.

Results: R4-1 (= O knows (L wants to perform (getting Runoff working)) is activated).
Activation of Rule 4 deleted from WS.

Cycle 1-14 -- Match

Match R4-11 with H1 = Helpee wants to perform a task.

Results: R4-11 is an activation of H2.
L corresponds to the Helpee.
(Getting Runoff working) corresponds to the task.

Cycle 1-15 -- Match

Match R0-1 with left half of

```
RuleVa = If O knows (a person says
                 (he executes a process with an instrumenl)),
            then O knows (that person is saying
                 (he performs (the execution of the process)
                      with the instrument).
```

Results: R0-1 is an activation of the left half of RuleVa.

L corresponds to the person.
(getting Runoff working) corresponds to (... executes a process ...) .
How corresponds to the instrument (i.e., the means).
An activation of Rule Va is created in the WS.

Cycle 1-16 -- Deduce

Deduce applies RuleVa to RO-1.

Results: RVa-1 (= O knows (L asks (how do I perform (getting Runoff working)?)) is activated)..
Activation of Rule Va deleted from WS.

Cycle 1-17 -- Match

Match RVa-1 with Left half of

Rule2a = If O knows (a person asks how to perform a task),
then O knows (that person wants O to enable him
to perform that task).

Results: RVa-1 is an activation of the left half of Rule2a.
L corresponds to that person.
(L getting Runoff to work) corresponds to the task.
An activation of Rule 2a is created in the WS.

Cycle 1-18 -- Deduce

Deduce applies Rule2a to RVa-1

Results: R2-1 (= O knows (L wants O to enable him (L) to get Runoff working) is activated).
Activation of Rule 2a deleted from WS.

Cycle 1-19 -- Match

Match R2a-1 with H11 = Helpee wants Helper to enable him to to a task.

Results: O corresponds to Helper.
L corresponds to Helpee.
(L getting Runoff to work) corresponds to the task.

Cycle 1-20 -- Proteus

H1, H2 & H11 provide Proteus with enough evidence to create an activation of the Helping-Game.

Results: An activation of the Helping-game is created in the WS.

Cycle 1-21 -- Dialogue-game processor

The presence of an activation of a Dialogue-game in the WS leads to the calling of the processor specialized in this category of knowledge. The Dialogue Games Processor (DGP) makes use of a set of correspondences that have already been established by the matches which led to the activations of H1, H2, and H11:

Previous Results: L corresponds to Helpee
O corresponds to Helper
Case/3 (= (Runoff working)) corresponds to the task.

Once an activation of a game has led to the calling of the DGP, the processor accesses the entire collection of information about the game from the LTM representation of it. The items of knowledge in the game, with the particular parameters of this situation substituted appropriately, fall into one of three categories:

1. Already known to hearer (e.g. H1, H2 & H11). Items in this category are simply ignored, since it serves no purpose to re-assert them.

2. Contradict knowledge already held by the hearer (e.g., if O already knew, for sure, that L knew all about Runoff). If any item falls into this category, the hypothesis that this game is active is simply abandoned as inaccurate.

3. Items neither previously known or contradicted (the majority of the content of the typical case). In this case, the DGP creates activations of these items to represent the collection of implicit knowledge that follows from a recognition of the proposed game.

Results: Activations are created for all of the following:
H5 = L wants to be able to get (Runoff working) himself.
(being able to get (Runoff working) is a subgoal
to performing (Runoff working).)
H6 = O is able to enable L to get (Runoff working).
H8 = O is able to want to enable [i.e. is willing to enable]
L to get (Runoff working).
H10= L is permitted to get (Runoff working).

There is also a collection of knowledge within the game having to do with the conduct of the game, rather than what the participants need to successfully evoke it. These items of knowledge and goals are also established as activations by the DGP at this time:

Results: Activations are created for all of the following:
HGX1 = L knows of an execution scene (XS/HE).
HGX2 = L knows that if he perceives a particular
terminal state of this scene, this will
satisfy his wanting to perform the task.

HGX2C= (Thus) L wants tc perceive this terminal state
of XS/HE.

An ACTION/GOOD is an ACTION within the specification of
XS/HE which occurred in the past.

HGX3 = L knows that he has perceive the ACTION/GOOD.

HGX4 = L knows he expected to perceive it.

HGX5 = L wanted to perceive it.

An ACTION/BAD is an ACTION within the specification of
XS/HE which has not occurred in the past.

HGX6 = L knows he has not perceived the ACTION/BAD.

HGX7 = L knows he expected to perceive it.

HGX8 = L knows he wanted to perceive it.

(perceiving the ACTION/BAD is a subgoal to perceiving
the desired terminal state of XS/HE.)

HGX9 = L wants to describe the ACTION[s]/GOOD [to O].
(this describing is a subgoal to (O enables L to
perform the task)

HGX10= L wants to describe the ACTION[s]/BAD [to O].
(this describing is a subgoal to (O enables L to
perform the task)

Associated with processes, procedures, ceremonies, and the like, there may be an execution scene, which is in effect an abstract description of a complete performance of the object described. The execution scene resembles a flowchart, with the boxes being actions of one of the active agents involved.

In this case, the execution scene is for Runoff, a program which reads a file specified by the user, formats the contents of the file, and outputs this formatted material onto either the line printer or another file. The execution scene of Runoff, as stored in our model, is similar to figure A-1.

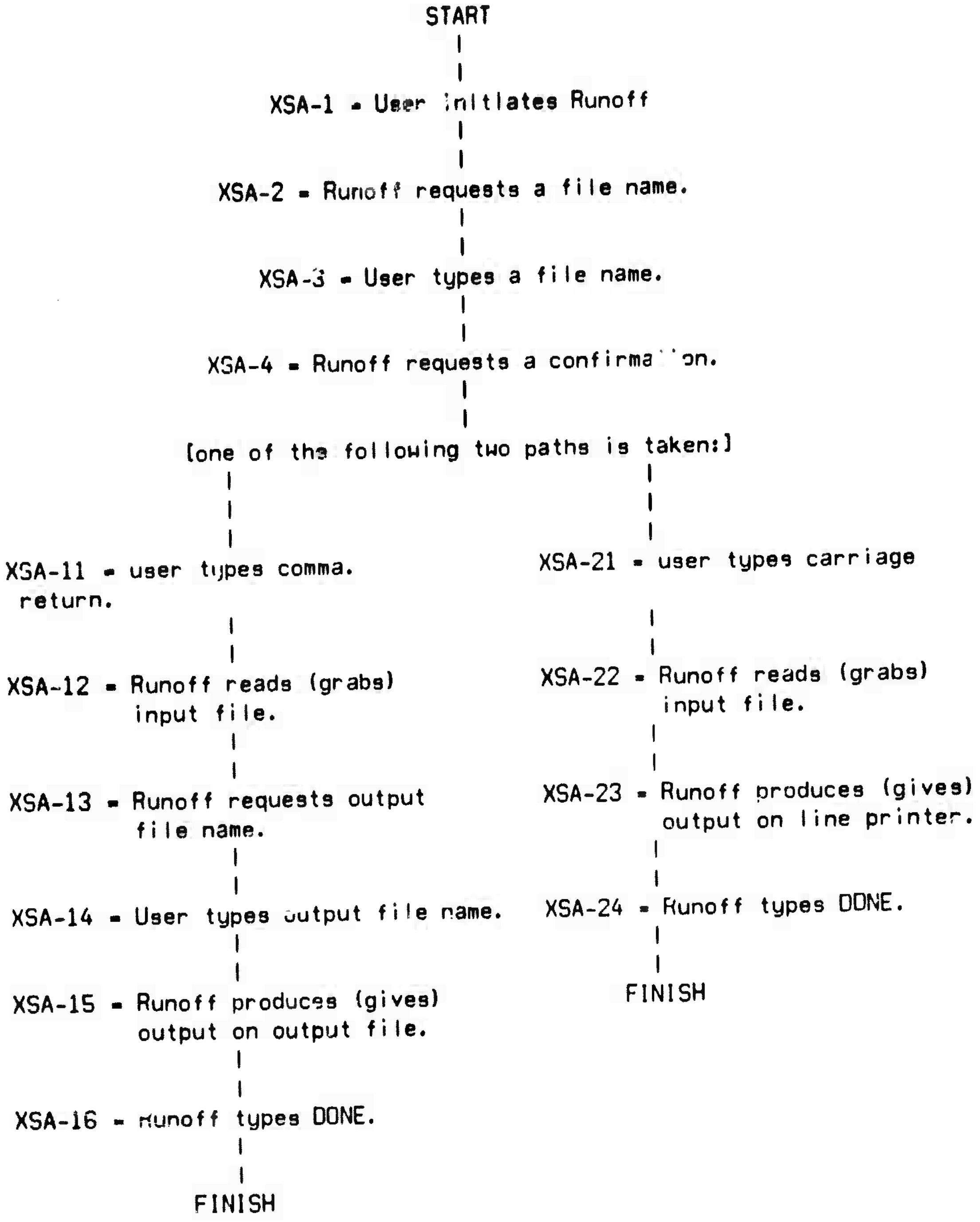
START
XSA-1 = User initiates Runoff
XSA-2 = Runoff requests a file name.
XSA-3 = User types a file name.
XSA-4 = Runoff requests a confirmation.
[one of the following two paths is taken:]
XSA-11 = user types comma.
return.
XSA-12 = Runoff reads (grabs) input file.
XSA-13 = Runoff requests output file name.
XSA-14 = User types output file name.
XSA-15 = Runoff produces (gives) output on output file.
XSA-16 = Runoff types DONE.
FINISH
XSA-21 = user types carriage
XSA-22 = Runoff reads (grabs) input file.
XSA-23 = Runoff produces (gives) output on line printer.
XSA-24 = Runoff types DONE.
FINISH

Figure A-1. XS/RO, THE RUNOFF EXECUTION SCENE.

Cycle 1-22 -- Proteus

As a result of the numerous references to Runoff and XS/HE, the activations for these two concepts are "highly active". Consequently, when Proteus is called, the concept XS/RO (the execution scene of the Runoff process) becomes active and, due to its similarity to XS/HE, is taken to be equivalent to it. Since XS/RO is more detailed (contains more information) than XS/HE, XS/RO is used in place if XS/HE in all of the expressions introduced in Cycle 1-21.

Something we passed over in the earlier examples was the issue of when the model is willing to stop processing a given piece of text and go on to the next one. It seems inappropriate to demand that the model wring all possible information and deductions out of each utterance. Yet there must be some demands made on the assimilation. An alternate form of the question is: what needs of his own does the hearer see the incoming text as potentially satisfying? We have taken the position that a hearer (tentatively) understands an utterance, when he successfully views it as serving some goal imputed to the speaker. That is, to a first approximation, the hearer has assimilated an utterance if he figures out why the speaker said it.

The model has already established (HGX9 and HGX10, above) that L wants to describe (implicitly, to O) certain actions in XS/RO that L expected to perceive, and in some cases, did. Thus, in the following utterances, we see the model matching the parsed input structure with one of these two goals, thus it is seen as being in service of a goal of the speaker, and need be examined no further (for the time being).

In the subsequent example, we use two new rules: RS (Satisfaction) and RQ (Quiescence). RS determines when an utterance is seen to satisfy a speaker's goal and RQ reacts to this detected satisfaction by marking the utterance quiescent. (Operationally, this means that in the next cycle, the Parser is called to input the next segment of text.)

We resume the example at the point where the first segment has been marked quiescent, and the Parser is called.

Cycle 2-1 -- Parser

Results: Case9a = O perceives that L declares (I executed it).

Cycle 2-2 -- I-processor

Results: Case9a = O perceives that L declares (L executed it).

Cycle 2-3 -- It-processor

The case frame associated with the concept "execute" specifies that the object concept is to be a process. The It-processor determines this and examines the WS to see if there is any active concept which is a process. In this case, there is only one: Runoff. Since this case is so clearly unambiguous, this simple-minded resolution scheme is adequate to the task. (We have outlines for more ambitious resolution schemes, but the dialogues we have examined have not yet required them.)

Results: Case9a = O perceives that L declares (L executed Runoff).

Cycles 2-4 & 2-5 -- Match and Deduce

As in cycles 1-3 and 1-4, Rule0 is used to transform "perceive" into "know".

Results: RO-1a = O knows that L declares (L executed Runoff).

Cycle 2-6 -- Match

Two items in the WS are matched to the two parts of the left half of RS:

RS = If a person knows a proposition
and
he knows that a second person wants that proposition,
then the first person knows that the realization of the
proposition satisfies the second person's desire for it.

Results: RO-1a = (O knows (L declares ...)) corresponds to
(a person knows a proposition)
O corresponds to the first person.
(L declares ...) corresponds to the proposition.
O knows HGX9 = (L want (L describe action/good))
corresponds to
(he knows the second person wants that proposition).
L corresponds to the second person.
(L describe action/good) corresponds to
the proposition.
(L declares (L executed Runoff)) corresponds to
(L describe action/good)
declare corresponds to describe
(L executed Runoff) corresponds to ((User initiate Runoff) past)
thus, (L executed Runoff) corresponds to action/good
An activation of RuleS is created in the WS.

Cycle 2-7 -- Deduce

Deduce applies RS to RO-1a and HGX9. Activation of Rule S deleted from WS.

Results: RS-1a (= O knows ((L declares ...) satisfies (L wants (L describe ...))) is activated).

Cycle 2-8 -- Match

Match RS-1a with left half of RQ.

RQ = If a person knows ((person2 utters something) satisfies
(person2 wants something else))
then the first person knows that he comprehends

(person2 uttering something) as constituting the
something else that person2 wanted.

Results: RS-1a corresponds to the left half of RQ.
O corresponds to the first person.
(person2 utters something) corresponds to
(L declares (L executed Runoff))
L corresponds to person 2
(L executed Runoff) corresponds to something.
(person2 wants something else) corresponds to
(L wants (L describe ...))
(L describe action/good) corresponds to something else.
An activation of RQ is created in the WS.
An activation of RQ is created in the WS.

Cycle 2-9 -- Deduce

Deduce applies RQ to RS-1a.

Results: RQ-1a = O knows (O comprehends
(L declare (L execute Runoff))
as constituting
(L describe action/good))
Activation of Rule Q deleted from WS.

Cycles 3-1 to 3-8

This set of cycles are exactly parallel to the preceding set. The structure implanted Into WS by the Parser is

Case/9b (= O perceives (L declares (it grabbed file/mine)))

The It-processor translates "it" to "Runoff". RuleO is used by Match and Deduce to replace "perceive" with "know". Match and Deduce then apply RS and RQ, to determine that Case/9b is comprehended as constituting another instance of (L describes action/good) [XSA-12 or XSA-22, Runoff reads (grabs) input file].

Cycles 4-1 to 4-8

Similarly, the Parser-produced structure:

Case/9c (= it said done)
is also found to be comprehended as constituting an instance of (L describes action/good) [XSA-16 or XSA-24, Runoff types DONE].

Cycles 5-1 to 5-10

A nearly identical sequence of cycles applies to the next Parser-input:

Case/9d (= O perceive L declare (It did not produce output),)

except an additional Match/Deduce cycle is needed to apply Rp:

Rp = If a person declares that something didn't happen,
then he is declaring he did not perceive it happen

In this case, however, we determine that Case/9d is comprehended as constituting an instance of (L wants (L describe action/bad)) [XSA-15 = Runoff produces output on output file -- or -- XSA-23 = Runoff produces output on line printer].

What we have seen, then, is the setting up of the expectations that the speaker will (i.e. wants to) describe some things that went right, and some that didn't. The presence of these expectations has enabled the assimilation of the last four utterances, leading to the model's awareness that for L, steps XSA-1, XSA-12 or -22, and XSA-16 or -24 all proceeded as expected, but that L didn't perceive Runoff producing any output. Mechanisms outside the scope of this example determine that XSA-15 (Runoff produces output on output file) was perceivable to L (had it occurred), but that XSA-23 (Runoff produces output on the line printer) was not. This leads to the conclusion that XSA-23 probably was what had occurred, and thus to the subsequent explanation from O.

DISTRIBUTION LIST

AC-S, Education Programs
Education Center, MCDEC
Quantico, VA 22134

AFHRL-AS (Dr. G.A. Eckstrand)
Wright-Patterson AFB
Ohio 45433

ARI Field Unit - Leavenworth
P.O.Box 3122
Ft. Leavenworth, KS 66027

Advanced Research Projects Agency
Administrative Services
1400 Wilson Blvd.
Arlington, VA 22209
Attn: Ardella Holloway

Air University Library
AUL-LSE 76-443
Maxwell AFB, AL 36112

Prof. Earl A. Alluisi
Code 287
Dept. of Psychology
Old Dominion University
Norfolk, VA 23508

Dr. Daniel Alpert
Computer-Based Enducation
Research Laboratory
University of Illinois
Urbana, IL 61801

Dr. John R. Anderson
Dept. of Psychology
Yale University
New Haven, CT 06520

Armed Forces Staff College
Norfolk, VA 23511
Attn: Library

Ms. Carole A. Bagley
Applications Analyst
Minnesota Educational
Computing Consortium
1925 Sather Ave.
Laudordale, MN 55113

Dr. James Baker
U.S. Army Research Institute
1300 Wilson Blvd.
Arlington, VA 22209

Dr. M.A. Bertin, Scientific Director
Office of Naval Research
Scientific Liaison Group-Tokyo
American Embassy
APO San Francisco 96503

Dr. Jack R. Borsting
Provost & Academic Dean
U.S. Naval Postgraduate School
Monterey, CA 93940

Dr. John Brackett
SofTech
460 Totten Pond Rd.
Waltham, MA 02154

Dr. Robert I. Branson
1A Tully Bldg.
Florida State University
Tallahassee, FL 32306

Dr. John Seeley Brown
Bolt, Beranek, and Newman, Inc.
50 Moulton St.
Cambridge, MA 02138

Dr. Victor Bunderson
Institute for Computer Uses
in Education
355 EDLC Brigham Young University
Provo, UT 84601

Dr. Ronald P. Carver
School of Education
University of Missouri-Kansas City
5100 Rockhill Rd.
Kansas City, MO 64110

Jacklyn Caselli
ERIC Clearinghouse on
Information Resources
Stanford University
School of Education - SCRDT
Stanford, CA 94305

Century Research Corporation
4113 Lee Highway
Arlington, VA 22207

Chairman, Leadership & Law Dept.
Div. of Professional Development
U.S. Naval Academy
Annapolis, MD 21402

Chief of Naval Education and
Training Support (01(A)
Pensacola, FL 32509

Dr. Kenneth E. Clark
College of Arts & Sciences
University of Rochester
River Campus Station
Rochester, NY 14627

Dr. Allan M. Collins
Bolt, Beranek and Newman, Inc.
50 Moulton St.
Cambridge, MA 02138

Dr. John J. Collins
Essex Corporation
6305 Caminito Estrellado
San Diego, CA 92120

Commandant
U.S. Army Institute of Administration
Attn: EA
Fort Benjamin Harrison, IN 46216

Commanding Officer
Naval Research Laboratory
Code 2627
Washington, DC 20390

Commanding Officer
Naval Health Research Center
San Diego, CA 92152
Attn: Library

Capt. H.J. Connery, USN
Navy Medical R&D Command
NNMC, Bethesda, MD 20014

Dr. T.E. Cotterman
AFHRL-ASR
Wright Patterson AFB
Ohio 45433

Mr. Joseph J. Cowan, Chief
Psychological Research Branch
(G-P-1-62)
U.S. Coast Guard Headquarters
Washington, DC 20590

DCDR, USAADMINCEN
Bldg. #1, A310
Attn: AT21-OED Library
Ft. Benjamin Harrison, IN 46216

Dr. Ruth Day
Dept. of Psychology
Yale University
2 Hillhouse Ave.
New Haven, CT 06520

Defense Documentation Center
Cameron Station, Bldg. 5
Alexandria, VA 22314
Attn: TC

Director, Office of Manpower
Utilization
HQ, Marine Corps (Code MPU)
BCB, Building 2009
Quantico, VA 22134

Director, Management Information
Systems Office
OSD, M&RA
Room 3B917, The Pentagon
Washington, DC 20301

Dr. Donald Dansereau
Dept. of Psychology
Texas Christian University
Fort Worth, TX 76129

Dr. Ralph Dusek
U.S. Army Research Institute
1300 Wilson Boulevard
Arlington, VA 22209

ERIC Facility-Acquisitions
4833 Rugby Ave.
Bethesda, MD 20014

Dr. John Eschenbrenner
McDonnell Douglas Astronautics
Company-East
PO Box 30204
St. Louis, MO 80230

Major I. N. Evonic
Canadian Forces Personnel
Aplied Research Unit
1107 Avenue Road
Toronto, Ontario, CANADA

Dr. Beatrice Farr
U.S. Army Research Institute
1300 Wilson Boulevard
Arlington, VA 22209

Dr. Marshall J. Farr, Director (4)
Personnel & Training Research Programs
Office of Navy Research (Code 458)
Arlington, VA 22217

Dr. James M. Ferstl
Employee Development:
Training Technologist
Bureau of Training
U.S. Civil Service Commission
Washington, DC 20415

Dr. Victor Fields
Dept. of Psychology
Montgomery College
Rockville, MD 20850

Dr. John Ford
Navy Personnel R&D Center
San Diego, CA 92152

Dr. Larry Francis
University of Illinois
Computer Based Educatonal
Research Laboratory
Champaign, Il 61801

Dr. John R. Frederiksen
Bolt, Beranek & Newman, Inc.
50 Moulton St.
Cambridge, MA 02138

Dr. Alfred R. Fregly
AFOSR-NL, Building 410
Bolling AFB, DC 20332

Dr. Frederick C. Frick
MIT Lincoln Laboratory
Room D 268
PO Box 73
Lexington, MA 02173

Dr. Vernon S. Gerlach
College of Education
146 Payne Bldg. B
Arizona State University
Tempe, AZ 85281

Dr. Robert Glaser
Co-Director
University of Pittsburgh
3939 O'Hara St.
Pittsburgh, PA 15213

Capt. D.M. Gragg, MC, USN
Head, Section on Medical Education
Uniformed Services Univ. of
the Health Sciences
6917 Arlington Road
Bethesda, MD 20014

Dr. Duncan Hansen
School of Education
Memphis State University
Memphis, TN 38118

Dr. Frank J. Harris
U.S. Army Research Institute
1300 Wilson Boulevard
Arlington, VA 22209

Col. Frank Hart, Director
Training Management Institute
U.S. Army, Bldg. 1725
Fort Eustis, VA 23604

Col. G.B. Howard
U.S. Army
Training Support Activity
Fort Eustis, VA 23604

HumRRO-Columbus Office
Suite 23
2601 Cross Country Dr.
Clumbus, GA 31906

HumRRO-Ft. Knox Office
P.O. Box 293
Fort Knox, KY 40121

HumRRO-Western Division
27857 Berwick Drive
Carmel, CA 93921
ATTN: Library

Human Resources Research Organizatio
400 Plaza Bldg.
Pace Blvd. at Fairfield Drive
Pensacola, FL 32505

Dr. Earl Hunt
Dept. of Psychology
University of Washington
Seattle, WA 98105

Instructional Technology Branch
AFHRL
Lowry AFB, CO 80230

Dr. Edgar Johnson
U.S. Army Research Institute
1300 Wilson Boulevard
Arlington, VA 22209

Dr. Wilson A. Judd
McDonnell-Douglas Astronautics Co.
East
Lowry AFB
Denver, CO 80230

Dr. Arnold F. Kanarick
Honeywell, Inc.
2600 Ridgeway Pkwy.
Minneapolis, MN 55413

Dr. Milton S. Katz, Chief
Individual Training & Performance
Evaluation Technical Area
U.S. Army Research Institute
1300 Wilson Boulevard
Arlington, VA 22209

Dr. Steven W. Keele
Dept. of Psychology
University of Oregon
Eugene, OR 97403

Dr. David Klahr
Dept. of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr. Alma E. Lantz
University of Denver
Denver Research Institute
Industrial Economics Division
Denver, CO 80210

Dr. Sylvia R. Mayer (MCIT)
HQ Electronic System Division
LG Hanscom Field
Bedford, MA 01730

William J. McLaurin
Room 301
Internal Revenue Service
2221 Jefferson Davis Hwy.
Arlington, VA 22202

Dr. Donald E. Meyer
U.S. Air Force
ATC-XPTD
Randolph AFB, TX 78148

Military Assistant for Human Resources
Office of the Director of Defense
Research & Engineering
Room 3D129, The Pentagon
Washington, DC 20301

Dr. Andrew R. Molnar
Science Education Dev. & Res.
National Science Foundation
Washington, DC 20550

Dr. Ross L. Morgan (AFHRL-ASR)
Wright-Patterson AFB
Ohio 45433

Dr. Leo Munday
Houghton Mifflin Co.
P.O. Box 1970
Iowa City, IA 52240

Navy Personnel R&D Center
Code 306
San Diego, CA 92152
Attn: Dr. James McGrath

Navy Personnel R&D Center
San Diego, CA 92152
Attn: Dr. J.D. Fletcher

Navy Personnel R&D Center
Code 1
San Diego, CA 92152
Attn: Dr. James McGrath

Navy Personnel R&D Center
San Diego, CA 92152
Attn: Library

Dr. Leon Nawrocki
U.S. Army Research Institute
1300 Wilson Boulevard
Arlington, VA 22209

Dr. Donald A. Norman
Dept. of Psychology C-009
University of California, San Diego
La Jolla, CA 92093

Dr. Harold F. O'Neil, Jr.
Advanced Research Projects Agency
Cybernetics Technology, Room 623
1400 Wilson Blvd.
Arlington, VA 22209

Mr. Thomas C. O'Sullivan
TRAC
1220 Sunset Plaza Dr.
Los Angeles, CA 90069

ONR Branch Office
495 Summer Street
Boston, MA 2210
Attn: Dr. James Lester

ONR Branch Office
1030 East Green Street
Pasadena, CA 91101
Attn: Dr. Eugene Gloye

ONR Branch Office
536 S. Clark Street
Chicago, IL 60605
Attn: Dr. Charles E. Davis

Office of Naval Research
Code 200
Arlington, VA 22217

Mr. Luigi Petrullo
2431 N. Edgewood St.
Arlington, VA 22207

Dr. Kenneth A. Polycyn
PCR Information Sciences Co.
Communication Satellite Applications
7600 Old Springhouse Rd.
McLean, VA 22101

Principal Civilian Advisor
for Education and Training
Naval Training Command, Code 00A
Pensacola, FL 32508
Attn: Dr. William L. Maloy

R.Dir. M. Rauch
P II 4
Bundesministerium der Verteidigung
Postfach 161
53 Bonn 1, GERMANY

Research Branch
AFMPC-DPMYP
Randolph AFB, TX 78148

Dr. Joseph W. Rigney
University of So. California
Behavioral Technology Laboratories
3717 S. Grand
Los Angeles, CA 90007

Dr. Marty Rockway (AFHRL-TT)
Lowry AFB
Colorado 80230

Dr. Andrew M. Rose
American Institutes for Research 1055
Thomas Jefferson St. NW
Washington, DC 20007

Dr. Leonard L. Rosenbaum
Chairman
Dept. of Psychology
Montgomery College
Rockville, MD 20850

Dr. Worth Scanland
Chief of Naval Education & Training
NAS, Pensacola, FL 32508

Scientific Advisor to the Chief
of Naval Personnel (Pers Or)
Naval Bureau of Personnel
Room 4410, Arlington Annex
Washington, DC 20370

Dr. Robert J. Seidel
Instructional Technology Group, HumRRO
300 N. Washington St.
Alexandria, VA 22314

A.A. Sjoholm, Head Technical Support
Navy Personnel R&D Center
Code 201
San Diego, CA 92152

Dr. A. L. Slafkosky
Scientific Advisor (Code RD-1)
HQ, U.S. Marine Corps
Washington, DC 20380

Dr. Marshall S. Smith
Associate Director
NIE-OPEPA
National Institute of Education
Washington, DC 20208

Dr. Alfred F. Smode, Director
Training Analysis & Evaluation Group
Department of the Navy
Orlando, FL 32813

Dr. Richard Snow
Stanford University
School of Education
Stanford, CA 94305

LCDR J.W. Snyder, Jr.
F-14 Training Model Manager
VF-124
San Diego, CA 92025

Dr. William Strobie
McDonnell-Douglas Astronautics Co.
East
Lowry AFB
Denver, CO 80230

Dr. Persis Sturgis
Dept. of Psychology
California State University-Chico
Chico, CA 95926

Superintendent (Code 1424)
Naval Postgraduate School
Monterey, CA 93940

Technical Director
U.S. Army Research Institute for the
Behavioral & Social Sciences
1300 Wilson Boulevard
Arlington, VA 22209

Capt. Jack Thorpe, USAF
AFHRL-FTS
Williams AFB, AZ 85224

Mr. Walt W. Tornow
Control Data Corporation
Corporate Personnel Research
P.O. Box 0-HQN060
Minneapolis, MN 55440

Dr. Benton J. Underwood
Dept. of Psychology
Northwestern University
Evanston, IL 60201

Dr. Carl R. Vest
Battelle Memorial Institute
Washington Operations
2030 M Street NW
Washington, DC 20036

Dr. Joseph Ward
U.S. Army Research Institute
1300 Wilson Boulevard
Arlington, VA 22209

Dr. Claire E. Weinstein
Educational Psychology Dept.
University of Texas at Austin
Austin, TX 78712

Dr. David J. Weiss
Dept. of Psychology
N660 Elliott Hall
University of Minnesota
Minneapolis, MN 55455

Dr. Keith Wescourt
Dept. of Psychology
Stanford University
Stanford, CA 94305

Dr. Joseph L. Young
Director
Memory & Cognitive Processes
National Science Foundation
Washington, DC 20550

Robert Young
Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

122