American Journal of Computational Linguistics *Microfiche 35*

PROCEEDINGS

13TH ANNUAL MEETING

ASSOCIATION FOR COMPUTATIONAL LINGUISTICS

4: MODELING DISCOURSE AND WORLD KNOWLEDGE I

Timothy C. Diller, Editor

Sperry-Univac
St. Paul, Minnesota 55101

PREFACE

Session 4 centered around two major topics: modeling the flow of information in discourse and representing and utilizing the knowledge of the world shared by communicators. The paper by Deutsch describes a mechanism for identifying the referents of definite noun phrases within a task-oriented dialogue. (Note the closely related paper by Klappholz and Lockman in Session 5.) Bruce compares two discourse models: a "discourse grammar" which defines the set of found and/or likely discourse structures, and a "demand processor", which accounts for utterances as responses to and activators of internal demands. Phillips presents various cohesive links found in coherent discourse and then considers the inferential process essential to filling in knowledge only implicit in the linking mechanisms. Cullingford discusses the major components of SAM (Script Applier Mechanism), a computational system modeling the organization and management of extra-linguistic world knowledge. Badler describes a system for translating visual input into propositional descriptions of dscrete events. Focussing on a particular type of visual input (American Sign Language), Kegl and Chinchor present the use of frame analysis in describing various communicatory devices in ASL. Thanks to Carl Hewitt for chairing this session.--Timothy C. Diller, Program Committee Chairman

TABLE OF CONTENTS

American Journal of Computational Linguistics *Microfiche 35 : 4*

Establishing Context in Task-Oriented Dialogs

Barbara G. Deutsch

Artificial Intelligence Center
Stanford Research Institute
Menlo Park, California 94025

ABSTRACT

This paper describes part of the discourse component of a speech understanding system for task-oriented dialogs, specifically, a mechanism for establishing a focus of attention to aid in identifying the referents of definite noun phrases. In building a representation of the dialog context, the discourse processor takes advantage of the fact that task-oriented dialogs have a structure that closely parallels the structure of the task. The semantic network of the system is partitioned into focus spaces with each focus space containing only those concepts pertinent to the dialog relating to a subtask. The focus spaces are linked to their corresponding subtasks and ordered in a hierarchy determined by the relations among subtasks.

Acknowledgment

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and monitored by the U.S. Army Research Office under Contract No. DAHC04-75-C-0006.

INTRODUCTION

Language communication entails the transmission of concepts from the speaker's model of the world to the listener's. It is crucial that the speaker be able to communicate descriptions of concepts in his model in a way that allows the listener to pick out the relevant related concept in his model. In normal human communication it is not necessary to describe a concept in a completely unambiguous way. Contextual clues from both the situation and the surrounding dialog are counted on to help disambiguate. The listener's problem is to use that context to help in his identification of the concept being communicated. As a simple example, consider the utterance, "Hand me the box-end wrench," as it might occur in a conversation between two people working on a maintenance task. Although many box-end wrenches may be known to both the speaker and the listener, the fact that the listener has a particular box-end wrench in his hand makes the noun phrase unambiguous. (For other examples, see Norman, Rumelhart, et al., 1975). In the most extreme case, the use of pronouns depends entirely on the dialog context to determine the intended referent; "it" can refer to any single inanimate object or event.

A related problem arises with elliptical expressions. Often the surrounding dialog supplies enough information so that only a word or two suffices to communicate an entire (complex) idea. For example, consider the following exchange:

E: Bolt the pump to the platform.
A: O.K.
E: What tools are you using [to bolt the pump to the platform].
A: My fingers [are the tools I am using ...]

The expressions in brackets indicate the full utterance that was meant by the partial utterance. The listener must fill in this information from the surrounding dialog.

This paper considers such phenomena as they occur in task-oriented dialogs. By task-oriented dialog we mean conversation directed toward the completion of some task. In particular, we will be concerned with a computer-based consultant task in which an apprentice technician communicates with a computer system about the repair of electromechanical devices. The understanding system must maintain models of the world and of the dialog to disambiguate references in the apprentice's speech.

DISCOURSE IN SPEECH UNDERSTANDING

In a speech understanding system, the discourse component is one of several sources of knowledge that must interact in interpreting an utterance (see Paxton and A. Robinson, 1975; J. Robinson, 1975). Because of the uncertainty in the acoustic signal, it is important that higher level sources of knowledge like discourse give advice to the system at early stages in the analysis. For this reason, in our current speech system, routines for identifying the referents of definite noun phrases are applied as soon as a possible noun phrase is identified rather than waiting for an interpretation of the entire

utterance. In essence, the procedure entails searching the recent context to find possible referents and returning a list of candidates.

Ellipsis and pronoun resolution require a more local context than the resolution of nonpronominal definite noun phrases (DNPs). A description of the processing for ellipsis and pronoun resolution is contained in the section "Discourse Analysis and Pragmatics" in Walker et al., 1975. In this paper we concentrate on mechanisms for resolving DNPs.

DEFINITE NOUN PHRASES

The problem of resolving DNPs is basically a problem of finding a matching structure in memory. In the case of a computer system with a semantic network knowledge base, the problem is that of finding the network structure corresponding to the structure of the noun phrase. The node that maps onto the head node of the parse structure representing the noun phrase is the concept being identified by the noun phrase. For example, if the knowledge base contains the nodes shown in Figure 1 (and there are no other nodes with e (element) or s (superset) arcs to wrenches), then either node W1 or node W3, but not W2, will match the phrase "the box-end wrench". Matching is not always so straightforward. For example, consider the situation portrayed in Figure 2. The ed, or delineating element, arc (see Hendrix, 1975a) links a node to delineating information about members of the class that node represents. B-E is a set of

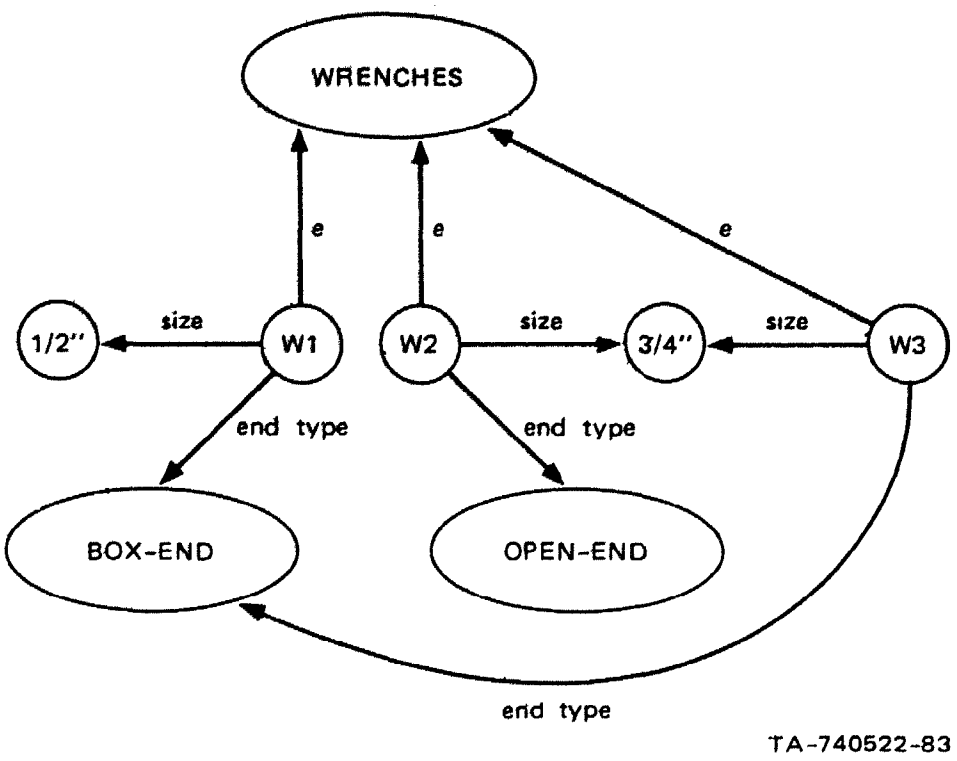


FIGURE 1 NETWORK DESCRIPTION OF THREE WRENCHES

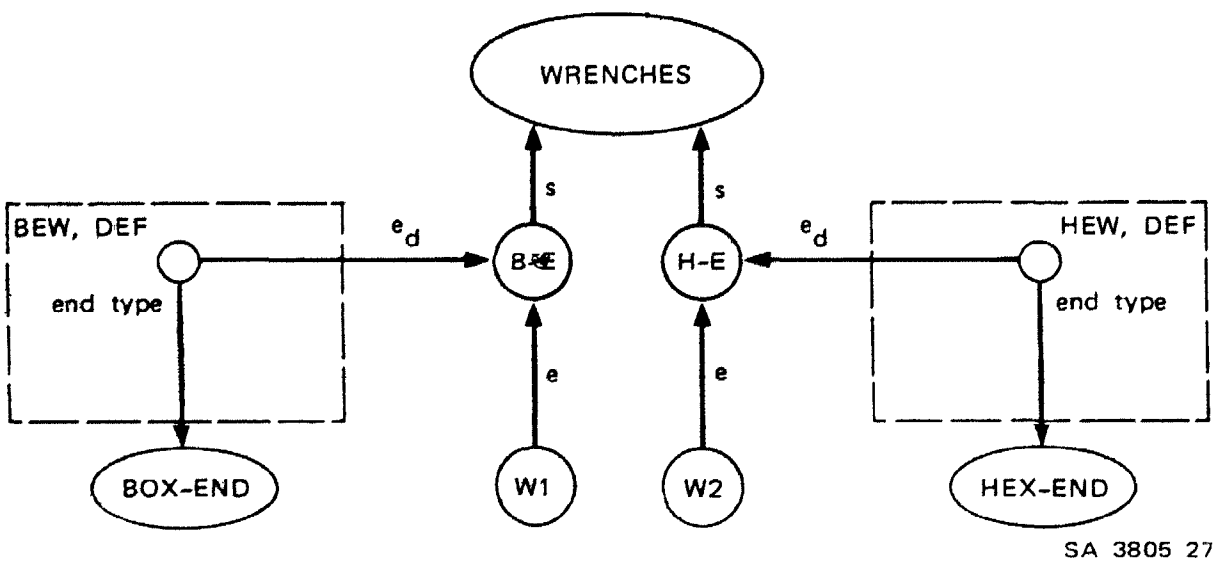


FIGURE 2 SEMANTIC NET SHOWING MEMBERS OF TWO SUBSETS OF THE SET "WRENCHES"

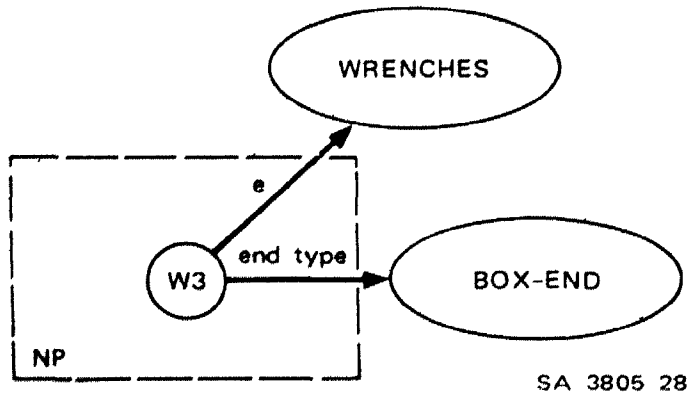


FIGURE 3 SEMANTIC NET SHOWING PARSE SPACE FOR "BOX-END WRENCH"

box-end wrenches to which W1 belongs. H-E is a set of hex-end wrenches to which W2 belongs. If the apprentice now says, "... the box-end wrench", he means W1. The utterance level structure created by parsing (see Hendrix, 1975b) for the phrase "the box-end wrench" is inside the space NP in Figure 3; some deduction must be done to establish the correspondence between W1 and W3.

The structure matching routines that form a basic part of the DNP resolver take as inputs a parse level network of nodes and arcs and a data network to match it against. (The current matcher was written by R. E. Fikes). In general, a large number of objects in the data net may be candidates for the matcher (i.e., objects that are elements of the same set as the object being identified by the DNP). Since, in itself, the matcher has no way of deciding which objects to consider first, additional mechanisms are needed to limit the search.

FOCUS SPACES

The discourse component must determine a subnet of the semantic net knowledge base for consideration by the matcher. That is, it must be able to establish as a local context that subset of the system's total knowledge base that is relevant at a given point in the dialog. This is analogous to determining what is in the user's focus of attention. Put another way, we would like to highlight certain nodes and arcs of the semantic network.

In task-oriented dialogs, the dialog context is actually a

composite of three different component contexts: a verbal context, a task context, and a context of general world knowledge. The verbal context includes the history of preceding utterances, their syntactic form, the objects and actions discussed in them, and the particular words used. The task context is the focus supplied by the task being worked on. It includes such information as: where the current subtask fits in the overall plan, what its subtasks are, what actions are likely to follow, what objects are important. The context of general world knowledge is the information that reflects a background understanding of the properties and interrelations of objects and actions: for example, the fact that tool boxes typically contain tools and that attaching entails some kind of fastening.

To highlight objects in the dialog and provide verbal context, network partitioning is used in a new way. Hendrix (1975a) has suggested imposing a logical partitioning on network structures for encoding logical connectives and quantifiers. Using the same technique, a focus partitioning may be used to divide the network into a number of local contexts. Nodes and arcs belong to both logical and focus spaces. The logical and focus partitions are independent of one another in the sense that the logical spaces on which a node or arc lies neither determine nor depend on the focus spaces in which the node or arc lies.

A new focus space is created for each subtask that enters the dialog. The task model (described shortly) imposes a hierarchical ordering, based on the subtask hierarchy, on these

spaces. This hierarchy determines what nodes and arcs are visible from a given space. The arcs and nodes that belong to a space are the only ones immediately visible from that space. Arcs and nodes in spaces that are above a given space in the hierarchy are potentially visible, but must be requested specifically to be seen. Other arcs and nodes are not visible.

A node may appear in any number of focus spaces. When the same object is used in two different subtasks, either the same or different aspects of the object may be in focus in the two subtasks. It is also possible for a node or arc to be in no focus space. In this case, the object is not strongly associated with the actual performance of any particular subtask. Such objects must be described relative to the global task environment. For completeness, we define a top-most space, called the "communal space", and a bottom-most space, called the "vista space". The communal space contains the relationships that are time invariant (e.g., the fact that tools are found in tool boxes) or common to all contexts. The vista space is below all other spaces and hence can see everything in the semantic net. This perspective is useful for determining all the relationships into which an object has entered.

The task model in our system will be embodied in a procedural net which encodes the task structure in a hierarchy of subtasks and encodes each subtask as a partial ordering of steps (Sacerdoti, 1975). The procedural net system also allows tasks to be expanded dynamically to further levels of detail when

necessary. A representation of the hierarchy of subtasks is important for reference resolution. An examination of task-oriented dialogs shows that references operate within tasks and up the hierarchy chain (Deutsch, 1974). Using the hierarchy of the procedural net to impose a hierarchy on the focus spaces enables us to search for references in hierarchical order. Having a representation of the partial ordering of tasks allows us to capture the alternatives the apprentice has in choosing subsequent tasks.

We have explicitly separated the three components of the dialog context. The representation of an object in a focus space will include only the relationships that have been mentioned in the dialog concerning the corresponding subtask or that are inherent in the procedural net description of the local task. Thus, the verbal component is supplied by the information recorded in the focus space hierarchy. Forward references to objects in the task (task component) are found by examining the procedural net. The general world knowledge component is information that is present in the communal space. When resolving a DNP, we can dynamically allocate effort between examining links in the local focus space, looking forward in the task, looking back up the focus space hierarchy, and looking deeper into knowledge base information.

GENERAL STRATEGY

The strategy we are currently exploring is first to examine

the currently active focus space and then to examine the next level of detail in the task. If the referent cannot be found in either of these locations, we look up the focus space hierarchy and then further down the task chain. The current context to be used by the discourse processor includes:

(1) A focus space containing the objects currently in focus
(2) A link to the associated node in the task model
(3) A type flag used in setting up expectations.

The type is necessary because there are subdialogs that do not directly reflect on the task structure. For example, there are subdialogs about tool identification ("What is a wheelpuller?") and tool use ("How do I use this wrench?"). References in these subdialogs do not follow the same focus space hierarchy and task structure.

The dialog shown in Table 1 will be examined to show how a combination of a task model and focus spaces may be used to help resolve DNPs.

E: I would like you to assemble the air compressor.
A: O.K.
E: I suggest you begin by attaching the pump to the platform.
A: O.K.
E: What are you doing now?
A: Using the pliers to get the nuts in underneath the platform.
E: I realize this is a difficult task.
A: I'm tightening the bolts now. They're all in place.
E: Good.
A: How tightly should I install this pipe elbow that fits into the pump?

Table 1: Subdialog for aircompressor assembly.

A partial procedural net for assembling an air compressor is

shown in Figure 4. The terms "install", "connect", "attach" refer to conceptual actions rather than lexical items. The dashed lines connect higher level tasks to their constituent subtasks. The time sequence of steps in the task is left to right. The partial ordering of tasks is encoded with the S and J nodes. The S, or ANDSPLIT, node indicates the beginning of parallel branches in the partial ordering. The nodes on arcs coming out of an S node may be done in any order. The J, or ANDJOIN, node indicates a point where several parallel tasks must be completed. The boxes labeled T are relevant to the subdialog fragment.

In the following analysis of the dialog, the utterances are considered sequentially. DNP resolution is considered in relation to the dialog history and the procedural net task model. (The search for references inside focus spaces is currently implemented; integration with the task model is not.) The context information listed under (1)-(3) above is shown in the accompanying figures as follows: (1) label on spaces in the network; (2) PNETTIE; (3) FSTYPE.

E: I would like you to assemble the air compressor.
A: O.K.
E: I suggest you begin by attaching the pump to the platform.

[At this point, we are at task T1; focus spaces FS0 and FS1 shown in Figure 5 have been set up.]

A: O.K.

[This could mean I'm done, but the response comes right after the instruction and the task takes a while.]

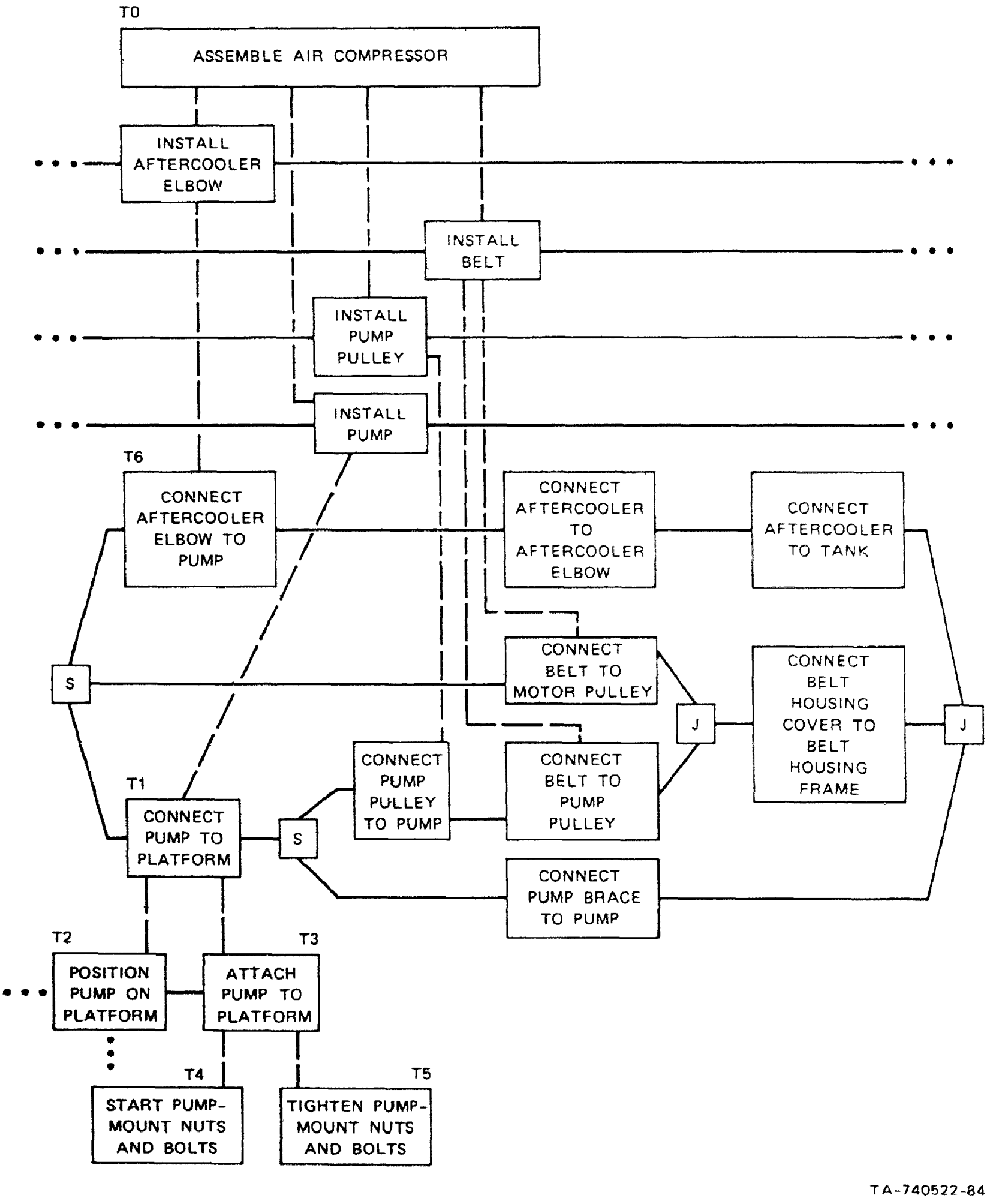


FIGURE 4 PARTIAL PROCEDURAL NET FOR ASSEMBLING AIR COMPRESSOR

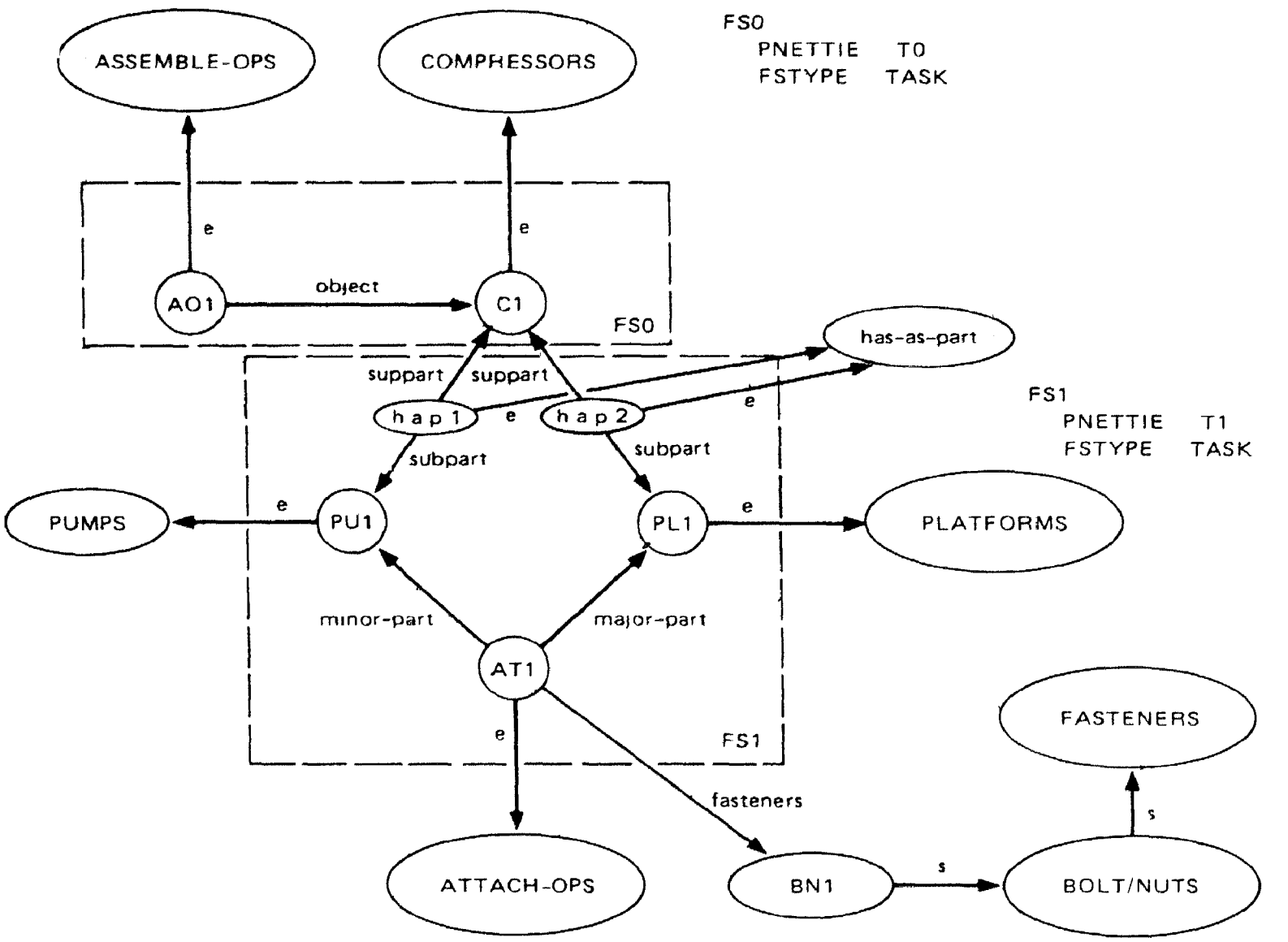


FIGURE 5 FOCUS SPACES FSO AND FS1

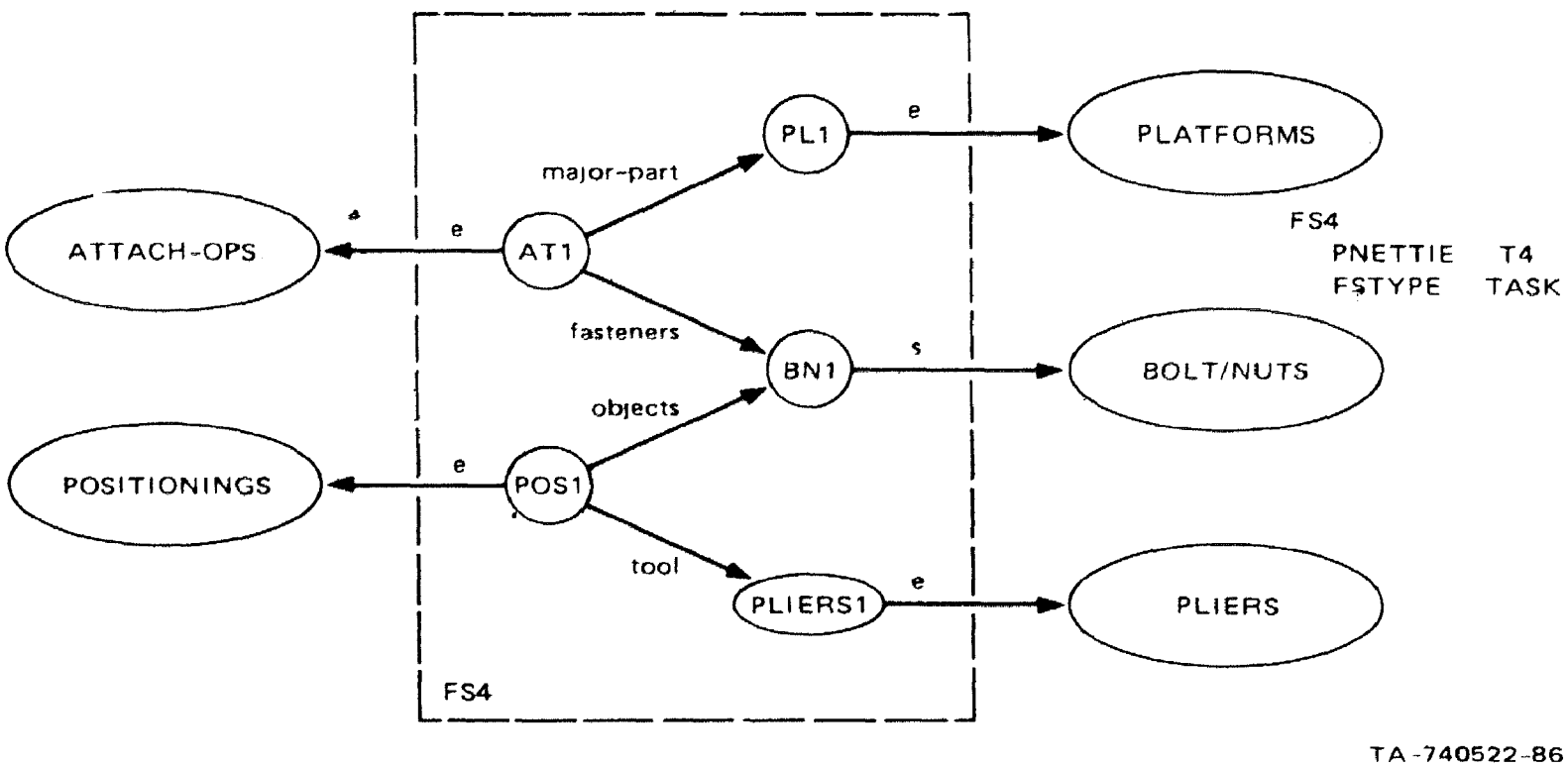


FIGURE 6 FOCUS SPACE FOR STARTING BOLT/NUTS OPERATION

E: What are you doing now?

[After a suitable waiting period, the expert queries the progress of the user.]

A: Using the pliers to get the nuts in underneath the platform.

["The pliers" can be resolved because there is only one pair; if this were not the case, the task model would have to be consulted. For both "the nuts" and "the platform", the FS hierarchy is consulted. "The platform", P1 is in focus in the current FS. There is no sign of nuts so we look forward in the task model. The relevant parts are located in subtask T4. This causes a new context, to be established as shown in Figure 6.]

E: I realize this is a difficult task.

[An attempt to assess the apprentice's perception of the problem. Note that at this point the task has barely begun and the expert does not have a very good model of the apprentice.]

A: I'm tightening the bolts now. They're all in place.

[FS4 contains "the bolts"; they were brought into focus when T4 was started. "They" is determined to refer to "the bolts" by checking the objects in the previous utterance for number agreement. Note that the last statement confirms the closure of T4. "Tighten" opens T5.]

E: Good.
A: How tightly should I install this pipe elbow that fits into the pump?

[There is no pipe elbow in the current FS. (Note that up until that point in the query the apprentice might have been asking about task T5). We close T5; because of the task structure this brings us back up to the top level. We are at the point of looking into new tasks. At present all of the tasks are considered equally. Eventually T6 is found to involve an elbow.]

In summation, then, the focus spaces provide a way of isolating certain parts of the semantic net, thus providing a way to focus on immediately relevant information. By tying the focus spaces to a model of the task, we are able to consider forward

task references. Both the task model and the focus spaces are linked to the general knowledge base; thus, it is possible to go from an item in either the task model or a focus space to other known but not previously referenced information about that item. The focus spaces and task model provide access to context information about objects in the domain, making it possible to focus on a relevant subset of the system's knowledge.

References

Deutsch, Barbara G. The Structure of Task-Oriented Dialogs. Contributed Papers, IEEE Symposium on Speech Recognition, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 15-19 April 1974. IEEE, New York. 1974, 250-254.

Hendrix, Gary G. Expanding the Utility of Semantic Networks Through Partitioning. Advance Papers of the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, Georgia, USSR, 3-8 September 1975, 115-121 (a).

Hendrix, Gary G. Semantic Processing for Speech Understanding. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975 (b).

Norman, D. A., Rumelhart, D. E., et al., Explorations in Cognition. W. H. Freeman and Company, San Francisco, 1975.

Paxton, William H., and Robinson, Ann E. System Integration and Control in a Speech Understanding System. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Robinson, Jane J. A Tuneable Performance Grammar. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Sacerdoti, Earl. A Structure for Plans and Behavior. Technical Note 109, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, August 1975.

Walker, Donald E., et al. Speech Understanding Research. Annual Report, Project 3804, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, June 1975.

American Journal of Computational Linguistics *Microfiche 35 : 19*

Discourse Models and Language Comprehension

Bertram C. Bruce

Bolt Beranek and Newman Inc.
50 Moulton Street, Cambridge, Massachusetts 02138

ABSTRACT

Higher order structures such as "discourse" and "intention" must be included in any complete theory of language understanding. This paper compares two approaches to modeling discourse. The first centers on the concept of a "discourse grammar" which defines the set of likely (i.e. easily understood) discourse structures.

A second approach is a "demand processing" model in which utterances create demands on both the speaker and the hearer. Responses to these demands are based on their relative "importance". The length of time they have been around, and conditions attached to each demand. The flow of responses provides another level of explanation for the discourse structure.

These two approaches are discussed in terms of flexibility, efficiency, and of their role in a more complete theory of discourse understanding.

*

1. Introduction

As has been said many times, understanding anything a problem, an action, a word - demands some knowledge of the context in which it appears. Certainly this is true of language, where an utterance s meaning may depend upon who the speaker is, when he is talking, what has just been said, who the listeners are, what the purpose of the conversation is, and so on. It is reasonable to define language understanding as the process of applying contextual knowledge to a sound (or string of symbols) to produce a change in that context. Successful language understanding occurs whenever the changes in the hearer s context (model of the world) coincides with changes the speaker intended.

Of course, stating a problem in a different way does not solve it. Instead it suggests a series of subsidiary questions such as:

(1) What is a context? What does it look like? What are its components, its structural characteristics?

(2) How does a new utterance change an existing context? What is the assimilation process? What must be kept; what can de discarded?

(3) How does a model of changing context account for observed phenomena such as the ability to switch contexts, and to return later (but not too much later)?

(4) How does the domain of conversation influence the

structure of a "context"? Do different mechanisms operate when the subject matter is tightly constrained?

It may be quite a while before questions of this type can be answered fully. This paper is a discussion of some of the issues and of the characteristics required of a solution. To do this, we will examine two classes of discourse models which have been proposed. The first is a "discourse grammar" approach which attempts to define the set of likely sequences of utterances. the second is a "demand processor" approach which attempts to account for utterances as responses to internal demands. Before discussing the specifics of these models, a sample dialogue, from the much restricted world of man-machine communication, is presented and discussed in Section 2.

2. An Example from Travel Budget Management

The problem of discourse understanding is complex enough that a complete analysis is certainly premature. We will examine here a micro-discourse, restricted to two conversants, one of them a machine, and, a conversation restricted in scope to that of managing a travel budget. Imagine that a person responsible for a travel budget is trying to decide how many people to send to an upcoming conference. He has available a computer system which can answer questions about the budget, conferences, trips, and so on. In talking to such a system, the travel budget

manager is engaging in an admittedly simple dialogue, but one which nevertheless exhibits many interesting properties of discourse understanding. (The following dialogue is hypothetical; however a system of this type is being developed at BBN (Woods, et al (1974, 1975))). The (human) manager begins:

(1) WHEN IS THE NEXT ASSP CONFERENCE?

May 15-18, 1975

(2) IS THAT IN OUR CURRENT BUDGET YEAR?

Yes.

(3) HOW MUCH MONEY DO WE HAVE LEFT?

$316.41 remains in the budget.

(4) WHAT IS THE COST OF THE ASSP CONFERENCE?

The registration fee is $25. Would you like me to estimate the cost of a trip to the conference?

(5) YES. FOR THREE PEOPLE.

Assuming they travel by air ...

Assuming each person spends three days in Philadelphia ...

Assuming a per day expense of $25 ...

The cost would be $371.50.

(6) WILL THE BUDGET COVER THAT?

No, the deficit would be $55.09.

(7) ASSUME THEY GO BY BUS.

The cost would be $327.40. The budget deficit would be $10.99.

(8) OK. SCHEDULE A TRIP FOR THREE PEOPLE BY BUS TO THE ASCP MEETING.

Done. That leaves $ -10.99 in the budget.

Even in a simple dialogue like the one above we see important discourse phenomena. For example, the manager's input (4) is ambiguous to the system. "Cost" could mean "registration fee" or "cost of a trip to the conference". The system has to ask about this ambiguity, producing an embedded question-answer pair within the original question-answer pair (cf. Schegloff s (1972) "insertion sequences"). However, there is more than a Q-1, Q-2, A-2, A-1 pattern in (4) and (5). The system does not just say, "Did you mean fee cost or 'trip cost'?". Assuming that "fee cost" is a reasonable interpretation, and useful to know besides, it simply states that fact, and asks about the other interpretation, which would demand more computation. A discourse model should account for this apparent awareness of computational difficulty, which is exhibited in human conversation and between a human and our idealized machine above.

Another phenomenon worth noting in this dialogue is the variation in detail and precision among the utterances. Sentence (8) is fairly precise and complete. Since alternatives have been considered to the trip he has decided upon it is important.

stress those aspects of the trip - "three people", "by bus" - which have been in question. On the other hand, sentence (3) is clearly elliptical. This is all right since the question is merely exploratory. Furthermore, the previous question insures that "money ... left" refers to money in the current budget. An adequate discourse model should account as well for our apparent ability to accommodate for the speech channel capacity, to minimize transmission errors through the use of redundancy and stress, and in general to attempt to optimize the communication.

One way to account for these and related phenomena is to postulate a discourse grammar. The grammar might say that part of a dialogue is a "question-answer" pair, and that it may be recursive in the sense that question-answer pairs may be embedded within it. This approach is discussed in the next section. A contrasting approach is to say that each utterance produces "demands" in the heads of the listeners. Responses to these demands may take the form of subsequent utterances. This latter model is discussed in Section 4.

3. Grammar Models of Discourse

Upon reading a dialogue like the example in Section 2, most of us readily form an opinion about its structure. In any dialogue we see this kind of structure: one person is asking another to do something; two people are arguing about politics, or discussing a novel. There is almost always a structure higher

than the individual sentences. In the example of Section 2, the travel budget manager seems to be entering into a "schedule a trip" dialogue. His question about a future conference is one of the cues to a bundle of information known by both him and the system about scheduling trips. Such a bundle has been variously referred to as a "frame" (Minsky (1975), Winograd (1975)), a "script" (Abelson (1975), Schank and Abelson (1975)), a "theme" (Phillips (1975)), a "story schema" (Rumelhart (1975)), and a "social action paradigm" (Bruce (1975a, 1975b)).

The information associated with scheduling a trip includes facts about dates and times, about the budget, about travel, about conferences, and so on. It also includes "plans", that is, time ordered structures of beliefs about achieving "goals". In this case, the goal is scheduling a trip to a conference. (See also Bruce and Schmidt (1974), Schmidt (1975)). One such partially instantiated plan might be -

1. Find out to which budget the trip should belong.
2. Determine how much is in the budget (budget).
3. Figure the cost of the trip (tripcost).
4. Decide whether (budget - tripcost) is acceptable.
5. If acceptable, schedule the trip and stop.
6. If not acceptable, determine if trip can be modified to be cheaper.
    a. If modifiable, go to 3.
    b. If not modifiable, stop.

The steps (1 - 6) above are ordered, though nothing is said about their relative lengths. Also, there are variants on the plan where the order might be changed, e.g. step 3 might come before step 2 in some other plan. The structure of such a plan, coupled with the by now commonplace observation that a discourse is structured, leads to the natural idea of representing a discourse by a grammar. Such a grammar may be large; it may be probabilistic; it may apply in only limited domains. Nevertheless it does give some idea of what to expect in a dialogue and may play a central role in language comprehension.

A portion of the grammar for our example dialogue is shown in Figure 1. This is an Augmented Transition Network Network (ATN) in which the arcs may refer to other networks (PUSH arcs), may signify direct transitions to other states (JUMP arcs), or may signify conclusion of the path (POP arcs). For example, in addition to this "SCHEDULE" network there is an "ENTER" network wherein the manager describes a new trip to be entered and the system asks him questions to complete the description.

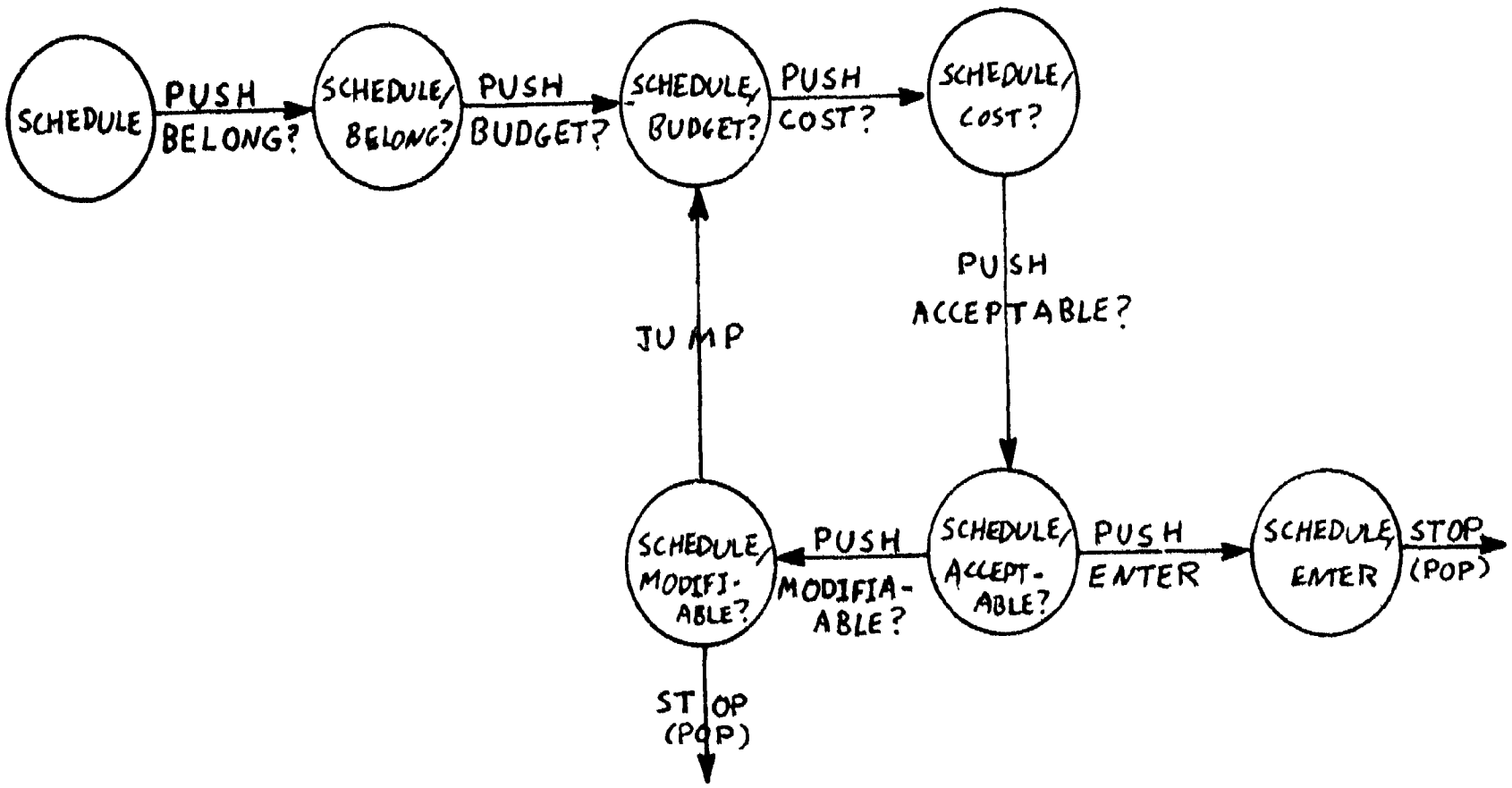


Fig. 1. ATN for scheduling a trip.

A discourse or dialogue grammar can be used with a modified ATN parser to "parse" a dialogue, generating both analyses of the current utterance and predictions about the one to come. In fact, one such modified parser and grammar has been implemented for the BBN speech system (Bruce(1975c), Woods, et al (1975)). For many dialogues, the grammar applies quite well, testing for the head verb in the utterance, the mood, and checking presuppositions of the action implied. When successful, it makes

corresponding predictions for application to the next utterance. Unfortunately, when the grammar fails it is not very good at recovering from its error.

Discourse grammars seem to be most effective in tightly constrained domains, more for instance in a discussion about how to cook a turkey, where there are specific subproblems to analyze, than in the travel budget management domain, and less still in a general question answering context. (Cf. Deutsch (1974, 1975)).

Lest it be thought that discourse parsing is just sentence parsing for "big sentences", I should emphasize some of the differences, differences which some would say preclude the use of terms like "grammar", "ATN", and "parsing". First, discourse parsing proceeds in a mode of partial parse, then output, then partial parse, etc. In other words, the goal is to derive information from the partial discourse which has occurred to suggest what may follow and to explicate the role of the current utterance. The parse is never completed, no structure is built. Since the entire discourse is not available to the parser (as the entire sentence is to a sentence parser), it is necessarily probabilistic. One can never know how the next utterance may alter the current interpretation of the trend of the dialogue. Another important difference is that PUSH's and POP's in the discourse grammar are "sloppy". That is, the participants in a dialogue may descend several levels ("Before you finish, let me

tell you about ...", "Before that ...") and never "pop" back up to the original level of the discourse. A discourse parser is faced with the peculiar phenomenon that a PUSH usually implies a POP but not always.

Some, but not all of these oddities of a discourse grammar are resolved by an approach which emphasizes internal models of the speaker and the listeners. This approach is discussed in the next section.

4. Demand Models of Discourse

One obvious characteristic of a discourse is that many processes may be occurring at once. A person cannot, nor does he wish to respond at one time to all unanswered questions, extend each unfinished line of thought, or deal with every inconsistency. While a grammar may predict the most likely action for a given point in a dialogue, it is not very good at suggesting alternatives out of the main line. There appears to be an additional mechanism of roughly the following form:

An event in a discourse (or prior to it) sets up a number of internal demands. Examples of such demands are to confirm what was said, explore its consequences, dispute it, answer it, etc. For any given event (such as an utterance) there may be none, one, or many demands created. A person's own action may place demands upon himself. If X asks a question of Y, then Y normally establishes an internal demand to answer the question. But X may

also establish a demand of the form, "check to see if the question has been answered". This latter demand may generate a later utterance such as, "Why haven t you answered me?".

Simple demand models already exist in a few systems. In general, they suggest that utterances are produced in response to conditions in the (internal model of the) environment rather than as units in a larger linguistic form. (See also Stansfield (1975)). It would be premature to argue that either a demand model or a grammar model is sufficient by itself. Instead, what follows is simply a description of a demand model for the travel budget management domain mentioned above.

Internal demands on the travel budget system help to explain how one computation of a response can be pushed down, while a whole dialogue takes place to obtain missing information, and how a computation can spawn subsequent expectations or digressions. Associated with each demand is a priority, a pointer (purpose) to the demand which spawned this one (if any), and a time marker indicating how long the demand has been around. An active unanswered question is a typical demand with high priority. Demands of lower priority include such things as a notice by the system that the manager is over his budget. Such a notice might not be communicated until after direct questions had been answered. The fact that some questions cannot be answered without more information leads to the

User-makes-query

System-asks-question

User-clarifies

System-answers-query

kind of embedding which is typically represented in a discourse grammar by a PUSH to a "clarification" state.

Counter-demands are questions the system has explicitly or implicitly asked the user. While it should not hold on to these as long as it does to demands, nor expect too strongly that they will be met, the system can reasonably expect that most counter-demands will be resolved in some way. This is an additional influence on the discourse structure.

A demand model also includes a representation of the current topic, the active focus of attention in the dialogue. For the travel budget system, it could be the actual budget, a hypothetical budget, a particular trip, or a conference. The current topic is used as an anchor point for resolving references and deciding how much detail to give in responses. Again, this structure leads to certain modes of interaction. For example, if the manager says "Enter a trip," the system notes that the current topic has changed to an incompletely described trip. This results in demands that cause standard fill-in questions to be asked. If the manager wants to complete the trip description later, then the completion of the trip description becomes a low

priority demand.

5. Synthesis?

Discourse has been an object of study for many both in and out of the field of computational linguistics. Especially worth noting is the work of sociolinguists such as Labov (1972), Sacks, Schegloff, and Jefferson (1975), and Schegloff (1972). Linguists (e.g. Grimes), sociologists (e.g. Goffman (1971)), and philosophers (e.g. Austin (1962), Searle (1969)) have important direct or related contributions. I certainly can't presume in this short paper to give the definitive solution to all the problems revolving around the discourse question. What I have tried to do is to emphasize a distinction in approach between looking at a discourse as a linguistic whole with subparts being individual utterances, and as a side effect of responses to task demands.

Both approaches are useful in exemplifying ways in which the otherwise hazy area of discourse might be modeled. The grammar approach makes the strongest statement about actual discourse structure and can best be used where the structure is well known or can be tightly constrained, e.g. in generating a discourse or in a man-machine system where the computer imposes control on the dialogue. A grammar and a discourse parser can be very efficient in such situations. When the dialogue is less predictable the (more bottom-up) demand processing approach may be more resistant

to "surprises" in the dialogue.

The ultimate discourse model probably contains aspects of both goal-directed grammars and of localized responses to demands. What should be particularly interesting to see is how characteristics of the model are affected by the type of discourse, human-machine v. human-human, problem-oriented v. information-exchanging, or new domain v. old.

REFERENCES

Abelson, Robert. "Concepts for Representing Mundane Reality in Plans". In Representation and Understanding: Studies in Cognitive Science (Ed: D. Bobrow and A. Collins), Academic Press, New York, 1975.

Austin, J. L. How to Do Things with Words. Clarendon Press, Oxford, 1962.

Bruce, Bertram. "Belief Systems and Language Understanding". BBN Report No. 2973, 1975a.

-----. "Generation as a Social Action". In Theoretical Issues in Natural Language Processing (Ed: B. L. Nash-Webber and R. C. Schank), ACL, 1975b.

-----. "Pragmatics in Speech Understanding". Proc. 4th IJCAI, Tbilisi, 1975c.

----- and C. F. Schmidt. "Episode Understanding and Belief Guided Parsing". Presented at 12th ACL Meeting, Amherst, 1974. (Also Rutgers Computer Science Dept. Report

CBM-TR-32).

Deutsch, Barbara G. "The Structure of Task Oriented Dialogues". Contributed Papers, IEEE Symposium on Speech Recognition, CMU, Pittsburgh, 1974.

-----. "Discourse Analysis and Pragmatics". In Speech Understanding Research (D. Walker, W. Paxton, J. Robinson, G. Hendrix, B. Deutsch, and A. Robinson), Annual Technical Report, SRI, 1975.

Goffman, Erving. Relations in Public. Basic Books, New York, 1971.

Grimes, Joseph. The Thread of Discourse. Mouton, Paris, in press.

Labov, William. "Rules for Ritual Insults". In Studies in Social Interaction (Ed: David Sudnow), The Free Press (Macmillan), 1972.

Minsky, Marvin. "A Framework for the Representation of Knowledge". In The Psychology of Computer Vision (Ed: P. Winston), 1975.

Phillips, Brian. Topic Analysis. Ph. D. Thesis, SUNY Buffalo, 1975.

Rumelhart, David. "Notes on a Schema for Stories". In Representation and Understanding: Studies in Cognitive Science (Ed: D. Bobrow and A. Collins), Academic Press, New York, 1975.

Sacks, Harvey, Emanuel Schegloff and Gail Jefferson. "A Simplest Systematics for the Organization of Turn-Taking for

Conversations". Semiotica, 1974.

Schank, Roger and Robert Abelson. "Scripts Plans and Knowledge". Proc. 4th IJCAI, Tbilisi, 1975.

Schegloff, Emanuel A. "Notes on a Conversational Practice: Formulating Place". In Studies in Social Interaction (Ed: David Sudnow), The Free Press (Macmillan), 1972.

Schmidt, Charles F. "Understanding Human Action: Recognizing the Motives and Plans of Other Persons". Carnegie Symposium on Cognition: Cognition and Social Behavior, CMU, Pittsburgh, 1975.

Searle, J. R. Speech Acts. Cambridge University Press, London, 1969.

Stansfield, James L. Programming a Dialogue Teaching Situation. Ph. D. Thesis, U. of Edinburgh, 1974.

Winograd, Terry. "Frame Representations and the Declarative-Procedural Controversy". In Representation and Understanding: Studies in Cognitive Science (Ed: D. Bobrow and A. Collins), Academic Press, New York, 1975.

Woods, William, M. Bates, B. Bruce, J. Colarusso, C. Cook, L. Gould, D. Grabel, J. Makhoul, B. Nash-Webber, R. Schwartz, J. Wolf. "Natural Communication with Computers, Final Report - Vol. I, Speech Understanding Research at BBN". BBN Report No. 2976, 1974.

Woods, William A., R. Schwartz, C. Cook, J. Klovstad, L. Bates, B. Nash-Webber, B. Bruce, J. Makhoul. "Speech Understanding Systems: QTPR 3". BBN Report No. 3115, 1975.

American Journal of Computational Linguistics *Microfiche 35 : 36*

JUDGING THE COHERENCY OF DISCOURSE

BRIAN PHILLIPS

Department of Information Engineering
University of Illinois at Chicago Circle
Box 4348, Chicago 60680

ABSTRACT

The component propositions of a coherent discourse exhibit anaphoric, spatio-temporal, causal and thematic structures. Not all of this structure is explicit, but must be inferred using a model of cognitive knowledge. The organization of knowledge in the model allows a bottom-up analysis of discourse. Further, knowledge is formed into small complexes rather than into the large monolithic structures found in Scripts/Frames.

1. The Structure of Coherent Discourse.

A discourse is judged coherent if its constituent propositions are connected. Various types of cohesive links are observed in discourse: anaphoric, spatial, temporal, causal and thematic. We will formally describe the structure of a well-formed discourse in terms of these connectives.

1.1 Anaphora.

Two kinds of anaphora can be distinguished. The first is marked by the presence of a proform (or by the repetition of a form):

(1) Henry travels too much. He is getting a foreign accent.

Antecedents may be nominal, verbal or clausal.

The second kind of anaphora has a dependent that is an abstract

term for the antecedent. For example,

(2) John put the car into 'reverse' instead of 'drive' and hit a wall. The mistake cost him $200 in repairs.

'Mistake' in (2) is an abstract characterization of the gear selection expressed in the first sentence.

A conventional way to label the recurring actors in discourse is as 'dramatis personae'. However cohesion can result not only from multiple appearances of people, but of any concept, as in (2).

1.2 Spatio-temporal and Causal Connectives.

Space, time and cause give coherency to a set of propositions.

(3) The King was in the counting house, counting out his money. The Queen was in the parlour, eating bread and honey.

The actions in (3) are set in different rooms, but of the same 'palace'.

(4) After Richard talked to the reporter, he went to lunch.

The temporal sequence of events in (4) is expressed by 'after'.

(5) John eats garlic. Martha avoids him.

To non-aficionados garlic is known only for its aroma, detection of which causes evasive action.

Cause, illustrated in (5) is an important discourse connective. Note however, that this is an ethnocentric view; in other cultures a different position may have to be taken, for example, a teleological world view (White: 1975).

This dimension of discourse structure is termed its 'plot' structure.

1.3 Thematicity.

Discourse is expected to have a theme, to have a topic. For example,

(6) Dino Frances drowned today in Middle Branch Resevoir after rescuing his son Dino Jr. who had fallen into the water while on a fishing trip.

is a new story from the New York Times, with a theme of, say, 'tragedy'.

Discourse may have more than one theme, but these should not conflict.

(7) Eating the fish made Gerry sick. He had measles in May.

In (7) we have an incoherent structure. The proposition 'Gerry sick' belongs both to a topic 'food-poisoning' and to a biography of illnesses. The analysis of fairy-tales by Lakoff (1972) suggests that discourse has a strictly tree-like thematic organization.

It is concluded that the propositions of a coherent discourse are connected either by coreference or (preferably) causally, and that it has a single theme (which may be the root of a tree of themes).

2. The Role of Inference.

Not all of discourse structure is overtly stated; discourse is highly elliptic. In (4) the discourse connective 'after' is present to mark a temporal sequence, but in (5) there is no realization of the causal relation between the two propositions. Normally one assumes that a discourse is coherent; hence (3) is most acceptable if the rooms are taken as being within the same habitation. Evidently a reader must infer omitted structure. The inferences are made from his cognitive store of world knowledge.

There is much discussion at present about inference as part of understanding. To make inferences is easy; the problem is to make the right ones. It helps to have a goal. It is suggested that discourse can be said to be understood when it has been judged coherent, as defined above.

3. Mechanisms of Inference.

A model of cognitive knowledge -- an encyclopedia -- should be capable of making the inferences necessary to form an opinion about the coherency of a discourse. The present encyclopedia originated with Hays (1973); a fuller description can be found in Phillips (1975). It is implemented as a directed graph. Labeled nodes characterize concepts and labeled arcs relations between concepts.

Propositions have a structure of case-related concepts, based on Fillmore (1969). This is our 'syntagmatic' organization of knowledge. As propositions are essentially the building blocks of discourse, we will not dwell on their structure here.

3.1 Anaphora.

If the dependent is a proform then part of understanding is to determine the correct antecedent. There are syntactic constraints (Langacker: 1969) which serve to narrow down choices for antecedents and to give an order of preference. The chosen antecedent will be the first that, when substituted for the proform, produces a meaningful proposition that is coherent in context.

A meaningful proposition is one that has a counterpart in the encyclopedia. The counterpart may be the self-same proposition, or more likely, a generalized proposition (hereafter a GP). For example, rather than 'Joan drink milk', we would expect to find 'animal imbibe liquid'.

How are GPs found? All concepts belong to partially ordered taxonomic structures in the encyclopedia (our 'paradigmatic' organization of concepts). From any concept it is possible to follow paradigmatic relations to a more general concept, which may be a constit-

uent of a proposition. An intersection of paradigmatic paths originating from each concept in a discourse proposition (hereafter a DP), taking account of syntagmatic structure, gives a GP. If there is no such intersection, then the DP is not consistent with encyclopedic knowledge.

Abstract terms can be defined by complexes of GPs, each having sufficient conceptual content to define situations in which they apply. For example, a definition of 'mistake' must be such that it applies to part of the first sentence in (2).

3.2 Space, Time and Cause.

To infer omitted spatio-temporal and causal relations (termed 'discursive' relations in the encyclopedia), it is also necessary to locate GPs. The encyclopedia, of course, includes these relations, but between GPs. Schematically, from a discourse proposition P_1 we can locate P_2, a GP, in the manner outlined above. P_2 may have a discursive relation R to another GP, P_3. A proposition P_4, a particularized version of P_3, and the relation R, between P_1 and P_4, can be added to the discourse, figure 1.

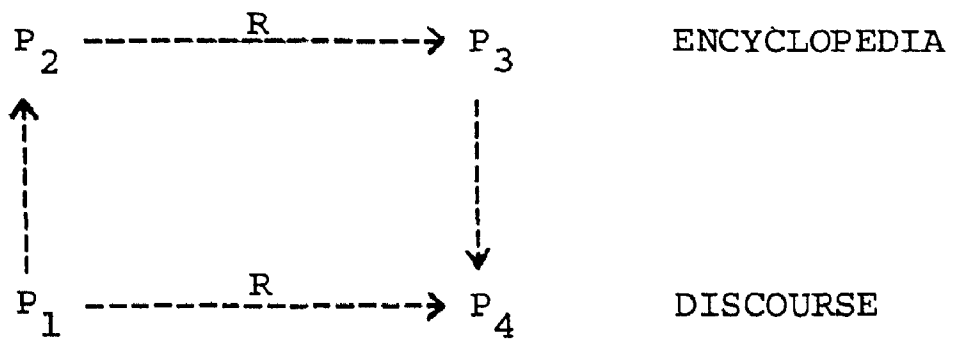


Figure 1.

Often P_4 will be a proposition already stated in the discourse; merely the relation need be inferred to augment the plot structure. It may, however, be necessary to infer a chain of propositions to link the original DPs. The question arises whether there is a limit on the number of propositions in a 'sensible' inferred path. Intuitively there is, but at present we have no formal insight.

3.3 Thematicity.

A theme is a complex of GPs, structurally indistinguishable from that used in characterizing abstract terms like 'mistake'. The potential presence of a theme is detected in the process of seeking GPs for DPs. All GPs, whether or not they are part of a thematic definition, can be located by paradigmatic searches; some GPs have additional structure indicating that they are components of themes. It is not sufficient to establish a theme for discourse by separately finding DPs that correspond to all the GPs of a theme. The thematic definition and the relevant part of the discourse must be tested holistically to ensure that the correct coreferentialities exist among the propositions.

3.4 Overview of Inference.

There are two basic processes underlying inference. First there is the process of locating a GP given a DP. This is implemented essentially by a breadth-first search through the paradigmatic structure of the encyclopedia. Secondly there is the process of matching a complex of propositions in discourse against an encyclopedic complex. The latter process is qualitatively different as it involves tests for coreference that the former does not.

Complexes of propositions have obvious functional similarities with 'Paraplates' (Wilks: 1975), 'Scripts' (Schank and Abelson: 1975) and

'Frames' (Minsky: 1975). Adding to the expanding terminology, our version is known as 'metalingual definitions'.

Metalingual definitions serve to define abstract terms ('mistake'), themes ('tragedy') and plans (used by Furugori (1974) in his robot planner). The distinctions are more terminological than substantive, their functions are interchangable; in other contexts a plan could be a theme, a theme an abstract term, etc.

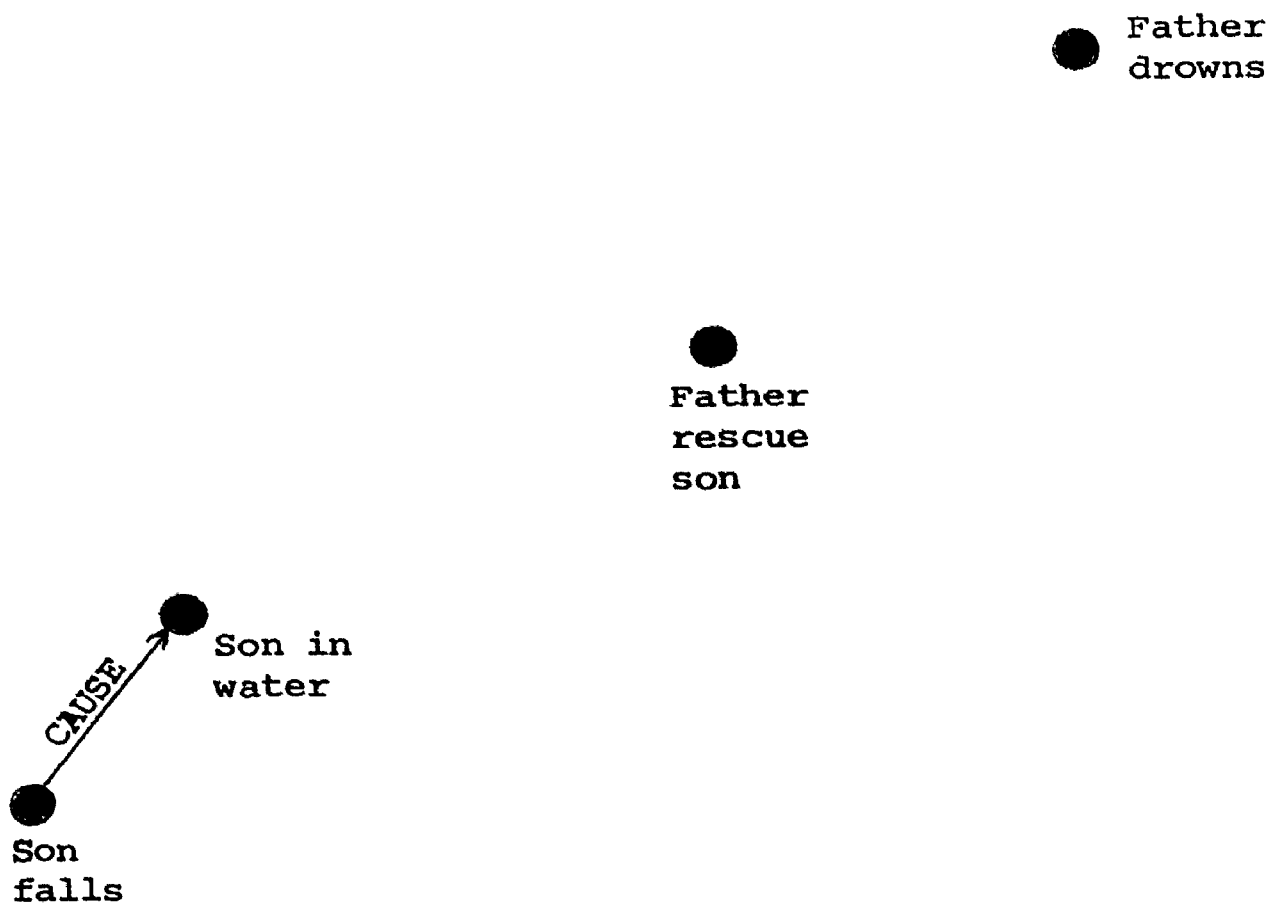
When an abstract concept has a metalingual definition, a matching discourse may be rewritten in terms of that concept. For example, 'buy' has such a definition, say '$person_1$ gives object to $person_2$, $person_2$ gives money to $person_1$'. To properly make the transduction to '$person_2$ buys object from $person_1$', there must be a case frame for 'buy' linked to concepts in its definition. A proposition produced by abstraction is structurally indistinguishable from a proposition that was in the original discourse, and can be subject to any encyclopedic process, including further abstraction. Conversely, if a proposition contains a concept having a metalingual definition, then the proposition can be decomposed into a complex of propositions patterned on the definition.

4. An Example.

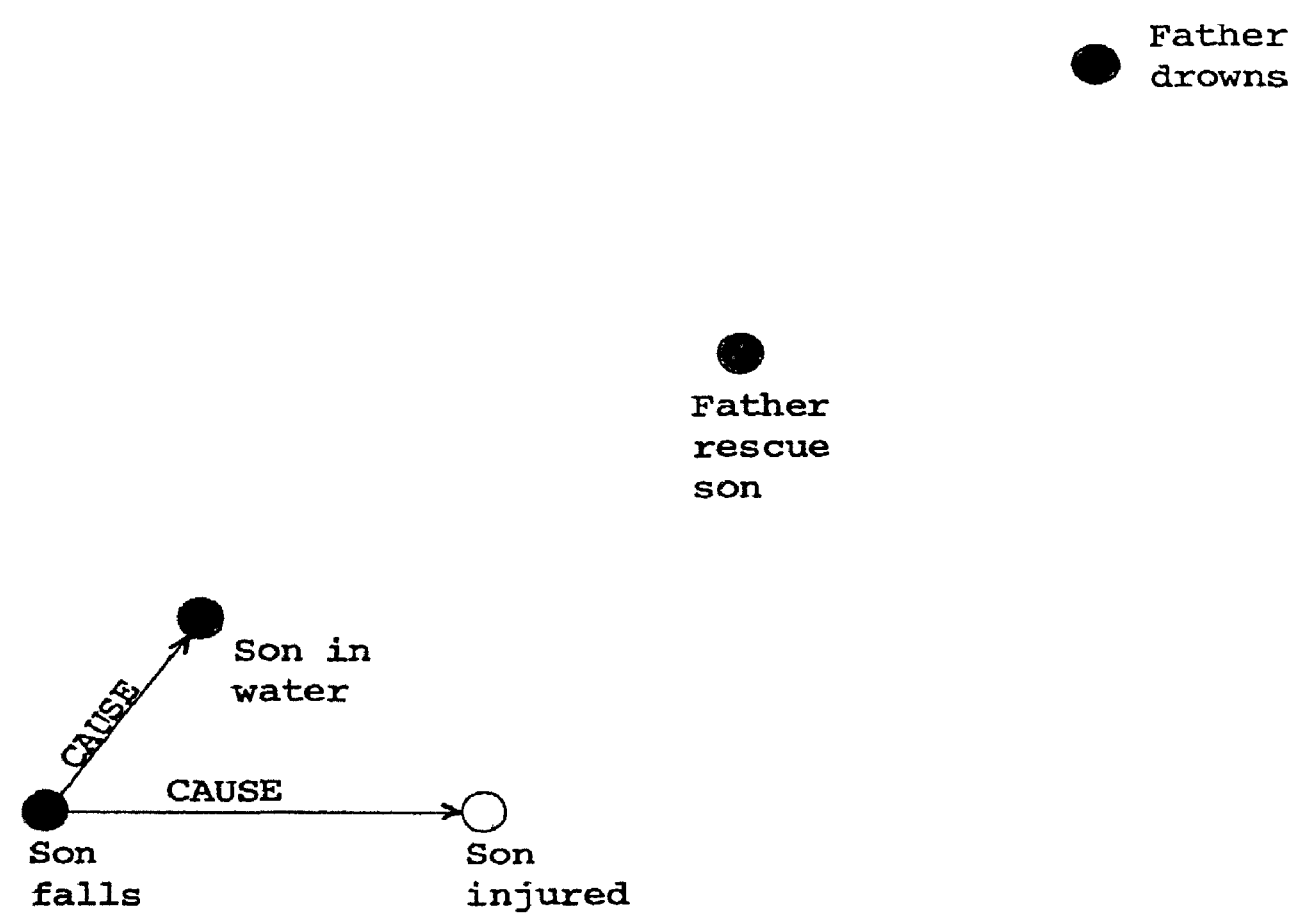
A schematic analysis of (6) shows the inference system in operation, resulting in a structure that satisfies the criteria of coherence.

At each step we will indicate the encyclopedic knowledge used in the inference, and the current state of the discourse. The original discourse propositions are indicated by ● and inferred propositions by ○

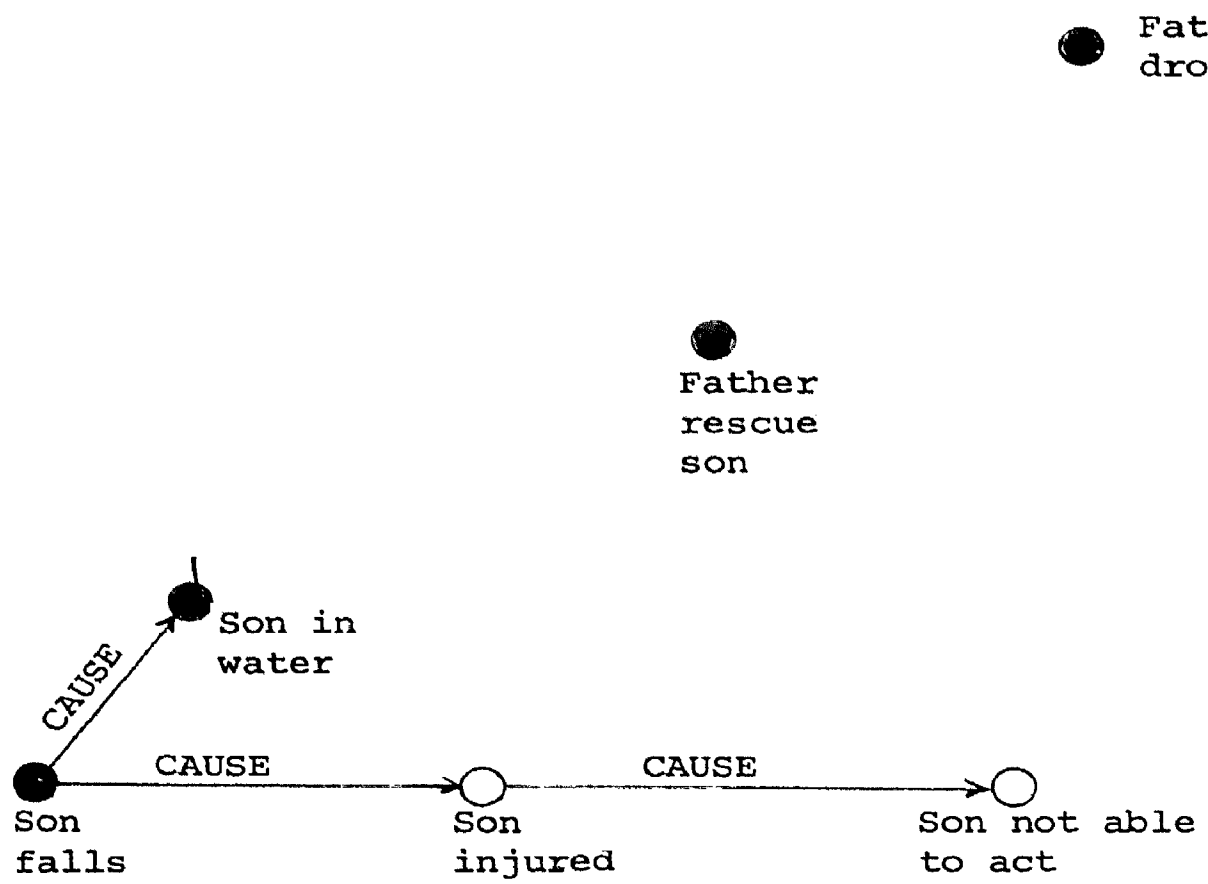
Step 0. Initial State.



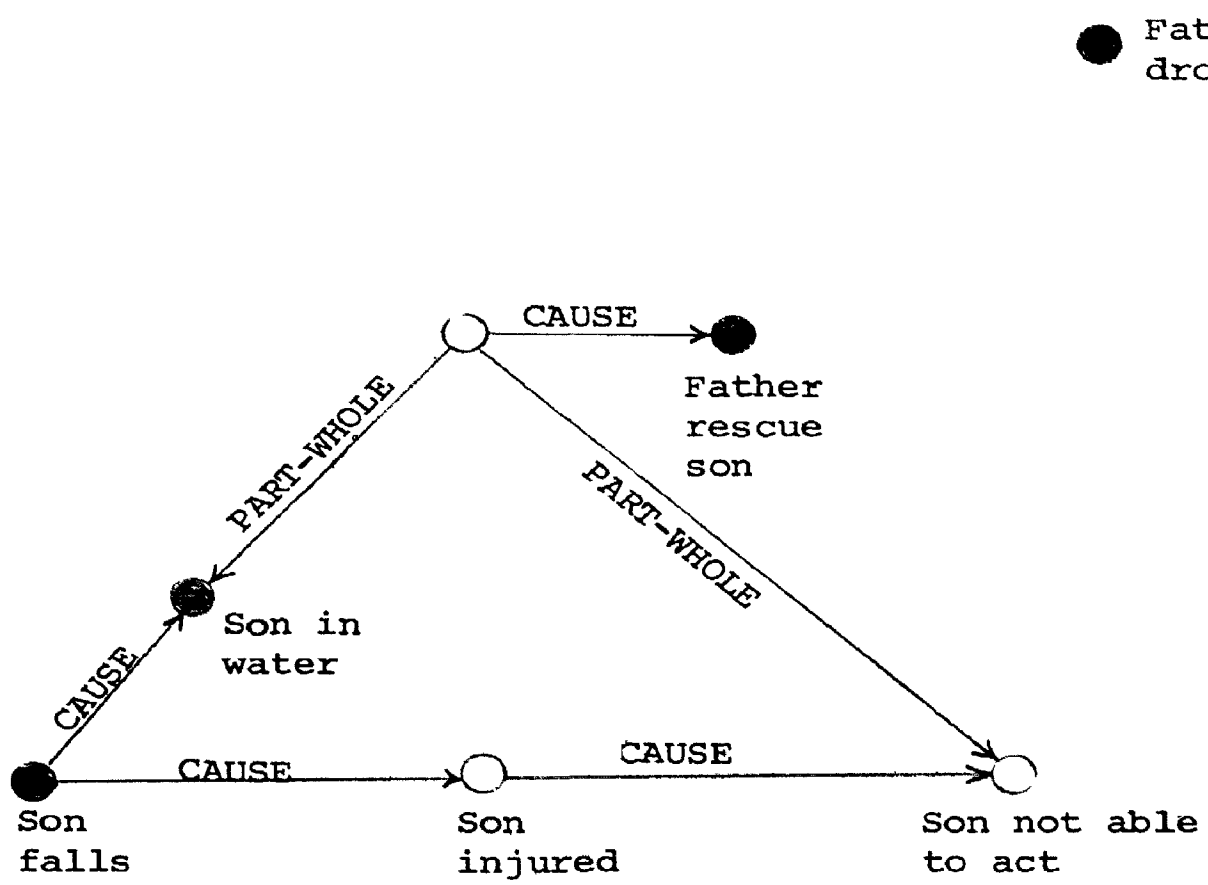
Step 1. Fall causes injury.



Step 2. Injury causes inability to act.

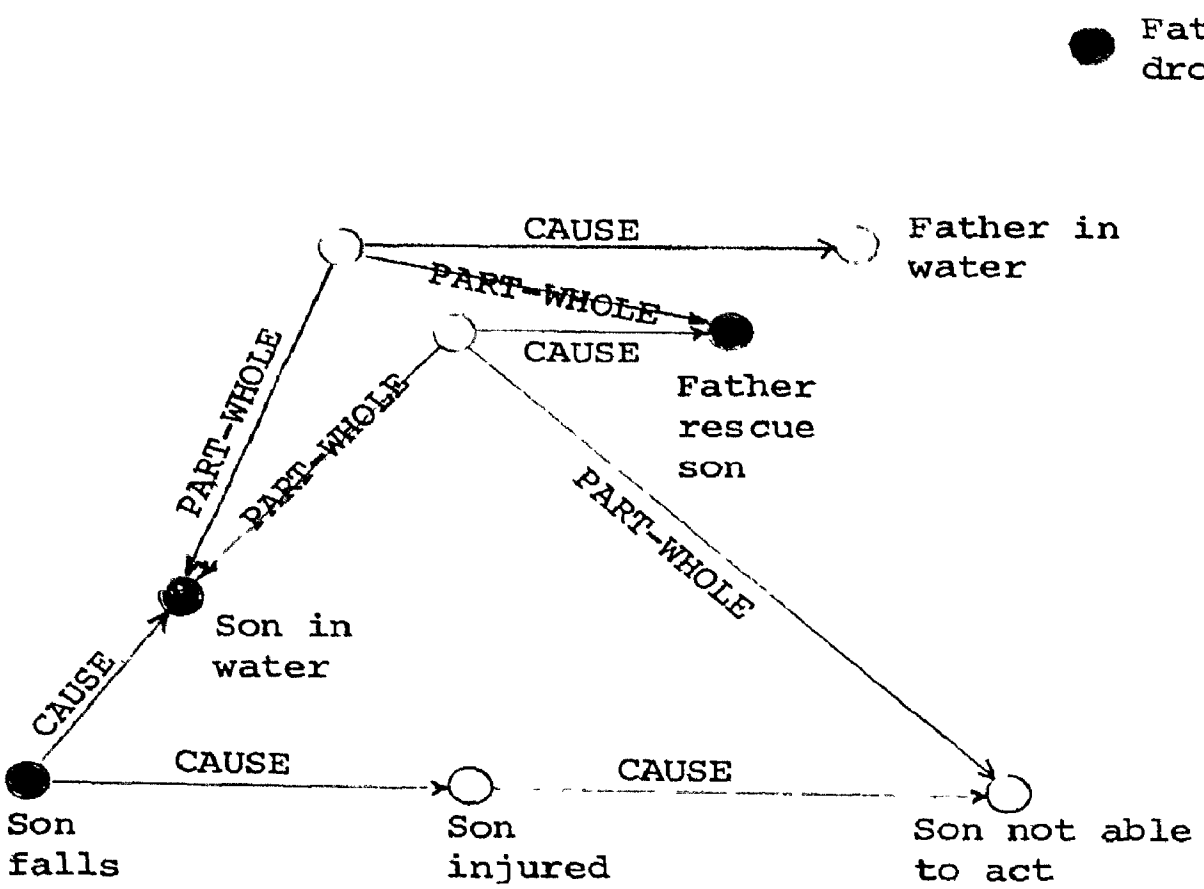


Step 3. In water and not able to act causes rescue.

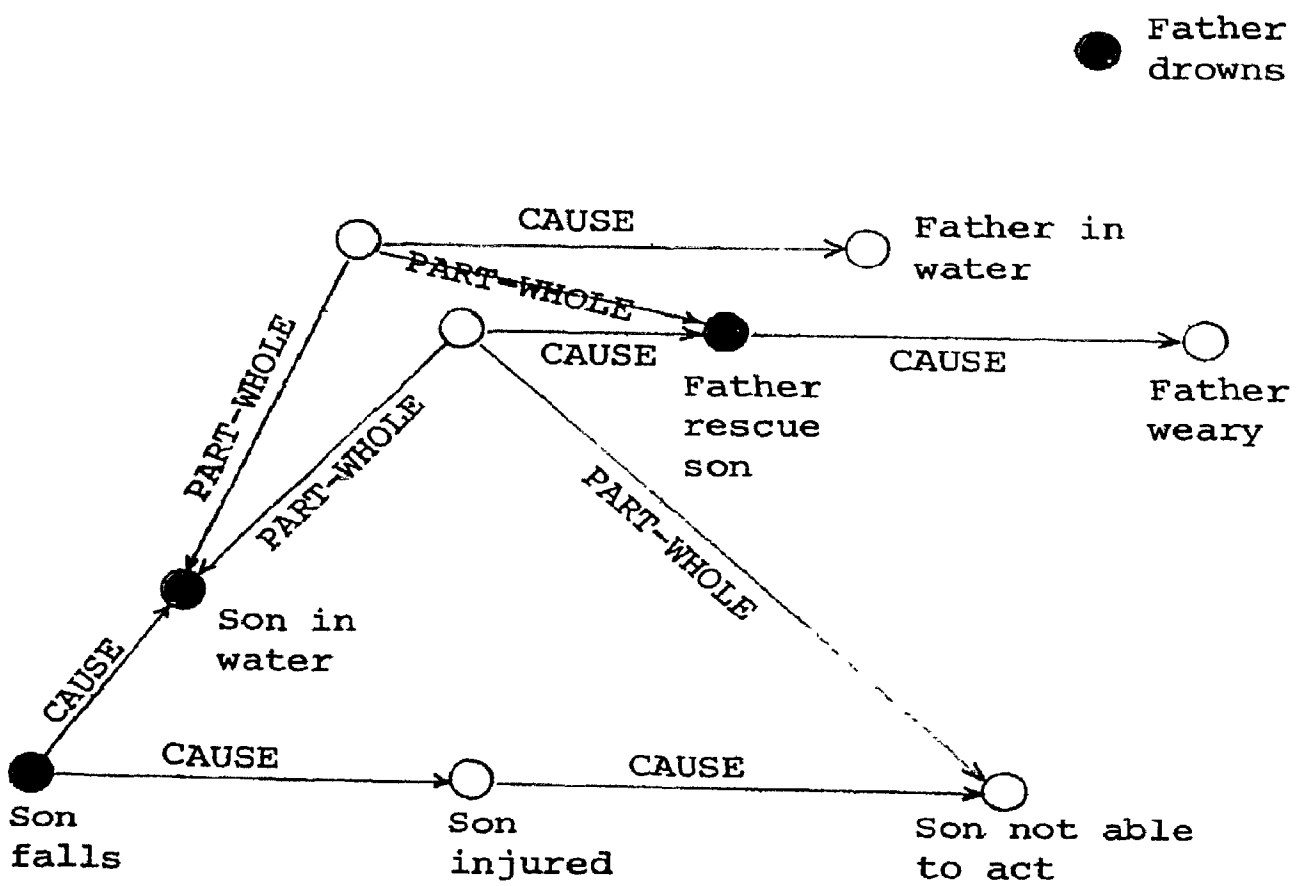


Conjunction is indicated by Part-whole relations. Note that a link to one of the original propositions has been established.

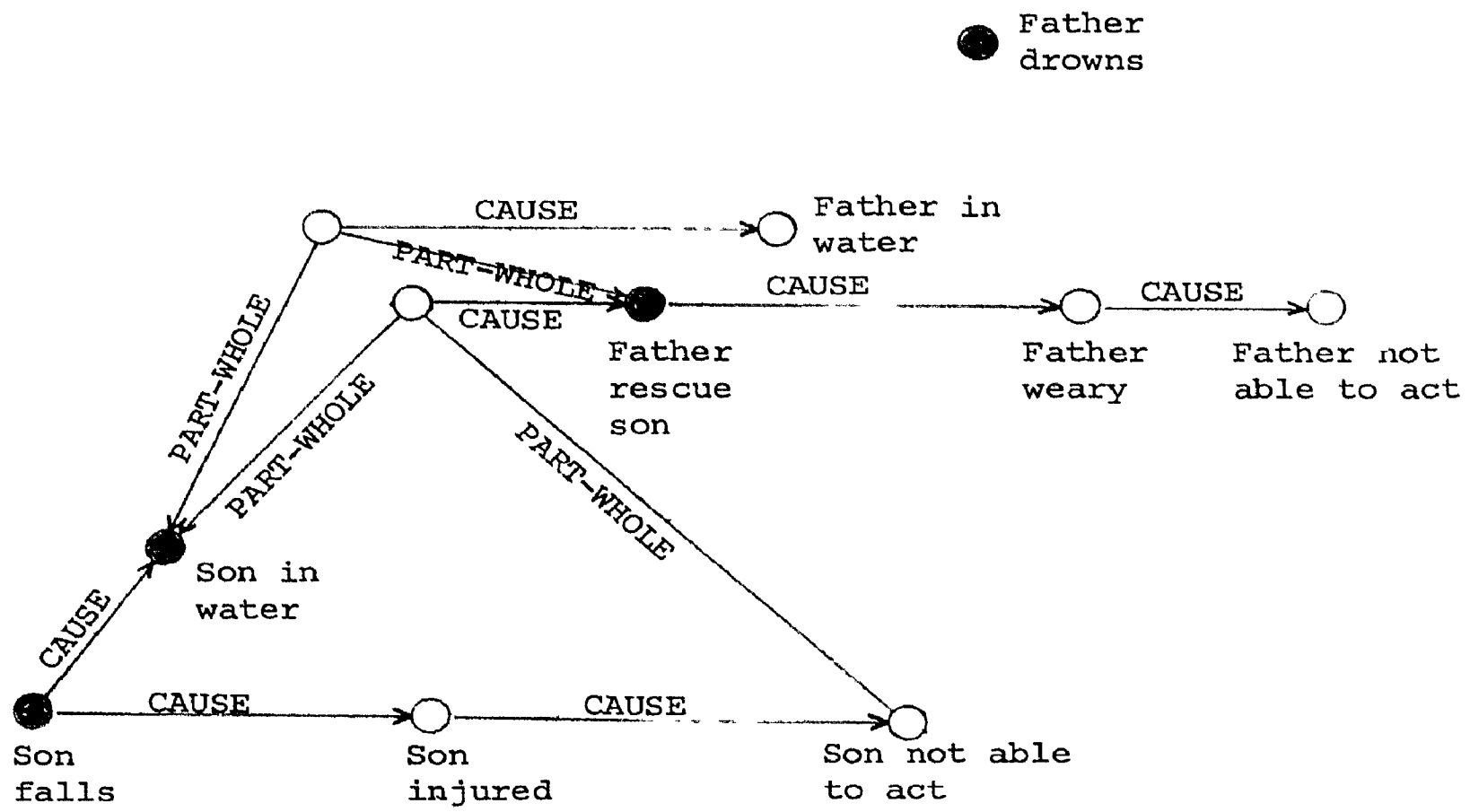
Step 4. To rescue someone who is in water it may be necessary to be in water.



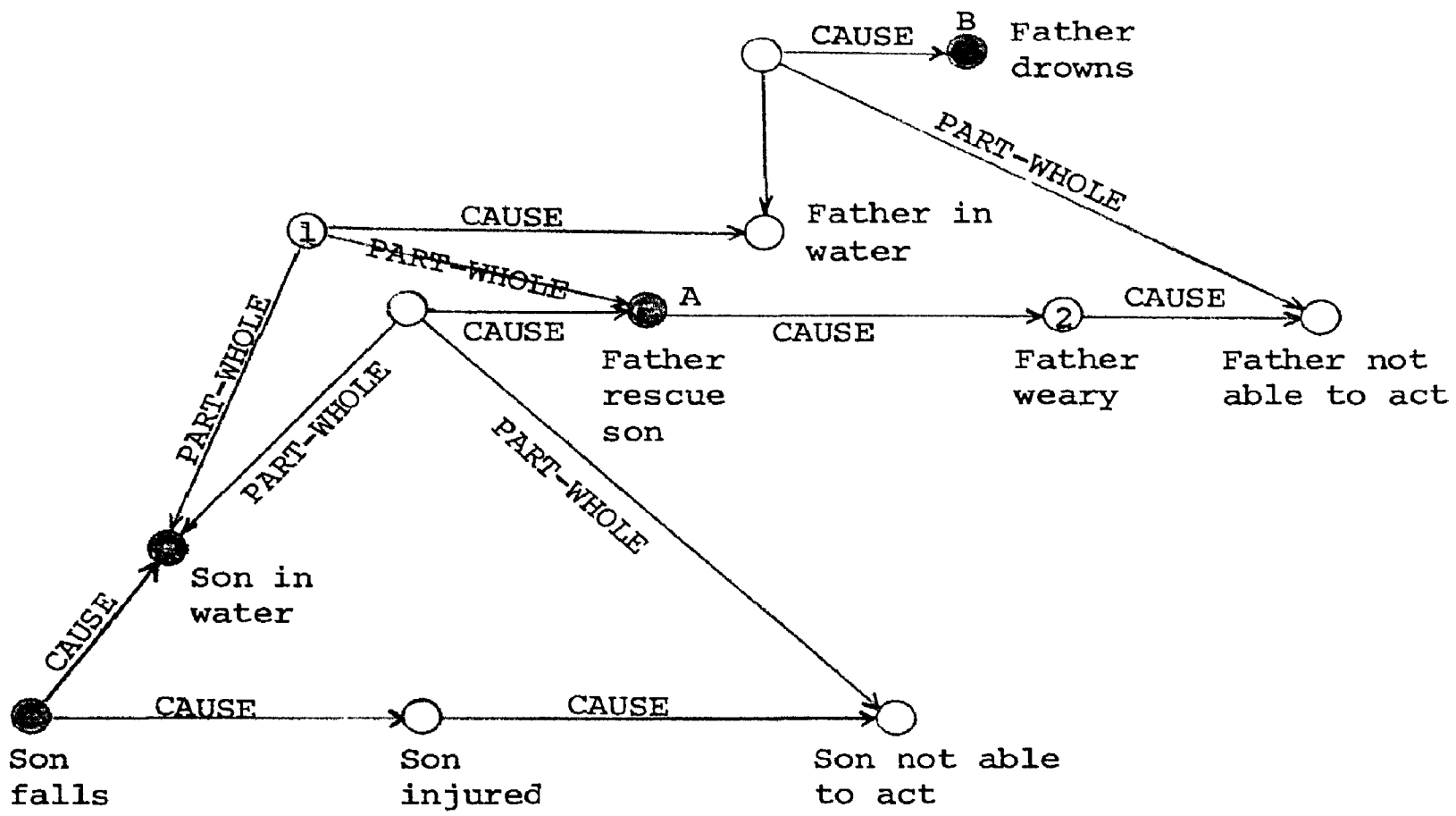
Step 5. Acting can make you weary.



Step 6. If weary then unable to act.



Step 7. If in water and not able to act then drown.



A link to the final proposition of the discourse is made. Corefer-

entiality conditions prevent 'son in water' and 'Father not able to act' conjoining to satisfy the conditions on this inference.

Note that the antecedent condition on this inference is the same as at step 3. Both resultant situations are possible, and are noted. The system can select either. However, the wrong choice does not lead to a connected structure, and a back up to the alternative has to be made.

The discourse now has an inferred causal structure connecting all the original propositions.

From a thematic analysis of drowning stories in general (Phillips: 1975), the common theme can be described as 'giving a cause for the person being in the water, and giving a cause for the victim not being able to act (thereby not being able to save himself)'. This theme fits the discourse by virtue of propositions ① and ②, which stand in causal relations to 'being in the water' and 'not able to act' for the victim. The theme 'tragedy' is defined as 'someone does something good and dies as a result of this action'. The father's rescue of his son and subsequent demise satisfy this theme (Ⓐ and Ⓑ). For the story to be coherent, these themes must not overlap; in fact we see that the 'drowning' theme is properly contained by 'tragedy'.

5. Discussion.

The analysis is so organized that the themes are determined in a bottom up manner, as are all generalized facts used in the analysis. Though not presently implemented, it should be possible to use potential themes, ones for which only some component propositions have been found, in a predictive manner.

The complexes of propositions, in metalingual definitions of themes and elsewhere, are really not that complex. The ones in the example contain only a few propositions. Each has only the essentials of the situation. The final structure arises from many small pieces of knowledge rather than from one monolithic aggregate. This seems to be a more natural organization, as each of the simpler structures can be freely applied in many contexts, rather than being bound to one situation.

The discourse judgement is relative to the knowledge of the hearer. Whether the inferences are those intended by the author is another question. Ideally they should be, or differences should be unimportant. A misleading inference indicates poor writing by the author; he has misjudged the knowledge of his audience.

Directing inferences on a discourse towards the goal of judging it coherent provides a normalized version of the discourse, if the process is successful. The normalized structure can form the basis for further processing: content analyis, stylistic analysis, etc. It may also provoke various questions, for example, we could ask if the inferences were correct; we have the 'rescue' situation applying to the father, but he wasn't rescued, why not.

References.

Fillmore, C. J. 1969. Toward a Modern Theory of Case.
In Reibel and Schane.

Furugori, T. 1974. "A memory model and simulation of memory processes for driving a car." Technical Report No. 77, Department of Computer Science, SUNY Buffalo.

Hays, D. G. 1973. Types of Processes on Cognitive Networks.
In Proceedings of the 1973 International Conference on Computational Linguistics. Pisa.

Lakoff, G. 1972. Structural Complexity in Fairy Tales.
The Study of Man 1, 128-150.

Langacker, R. W. 1969. On Pronominalization and the Chain of Command.
In Reibel and Schane.

Minsky, M. 1975. A Framework for Representing Knowledge.
In P. H. Winston (ed.), The Psychology of Computer Vision.
McGraw-Hill, NY.

Phillips, B. 1975. Topic Analysis. Unpublished Ph.D. Thesis.
SUNY Buffalo.

Reibel, D. A. and D. A. Schane (eds.). 1969. Modern Studies in English.
Readings in Transformational Grammar. Prentice-Hall,
Englewood Cliffs.

Schank, R. C. and R. P. Abelson. 1975. Scripts, Plans, and Knowledge.
In Advance Papers of the Fourth International Joint Conference on Artificial Intelligence. IJCAI.

White, M. 1975. Abstract Definition in the Cognitive Network:
The Metaphysical Terminology of a Contemporary Millenarian Community. Unpublished Ph.D. Thesis. SUNY Buffalo.

Wilks, Y. 1975. A Preferential, Pattern-Seeking, Semantics for
Natural Language Inference. Artificial Intelligence 6, 53-74.

American Journal of Computational Linguistics *Microfiche 35 : 50*

An Approach to the Organization of Mundane World Knowledge: The Generation and Management of Scripts

R. E. Cullingford

Yale University
New Haven, Connecticut 06511

ABSTRACT

In understanding stories or natural-language discourse, hearers draw upon an enormous base of shared world knowledge about common situations like going to restaurants, theaters or supermarkets to help establish the needed context. This paper presents an approach to the management of this type of knowledge based upon the concept of a situational script [Schank and Abelson, 1975]. The application of scripts in story understanding is illustrated via a computer model called SAM (Script Applier Mechanism).

In simple one-script stories, SAM constructs a trace through a preformed data structure containing the input, other events not mentioned but commonly assumed, the important

---

The research described in this paper was supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored under the Office of Naval Research under contract N00014-75-C-1111.

inferences associated with the events, and the interconnecting causal links. In more complicated stories, SAM handles the invocation and closing of parallel, nested and sequential scripts.

1.Ø Introduction

Natural-language processing research in recent years has increasingly focussed upon the modeling of human world knowledge and management of the resulting data base (1). This has come about largely because of the enormous problems encountered in the processing of texts, as opposed to single sentences, by traditional methods based upon syntactic analysis and low-level semantics. This state of affairs should not be surprising, since it is quite clear that people draw upon a huge store of shared, extra-linguistic world knowledge in understanding even the simplest stories or engaging in the most rudimentary conversation.

Much of the knowledge that hearers utilize to establish the background or context of a story appears to be episodic in nature, distilled from many experiences in common situations like going to restaurants, football games and supermarkets. This paper presents an approach to the representation and handling of this type of mundane world-knowledge based upon the concept of a situational script [Schank and Abelson, 1975]. The application

---

(1) See, for example, the emphasis on this area in "Theoretical Issues in Natural Language Processing", Proceedings of the Interdisciplinary Workshop in Computational Linguistics, 1Ø-13 June 1975, Cambridge, Mass.

of scripts in story understanding will be illustrated via a computer model called SAM (Script Applier Mechanism) devised for a PDP-10 computer at Yale University.

1.1 Examples of Story Understanding in SAM

Before attacking the various aspects of scripts and the particular activities carried on by SAM's script applier, it seems appropriate to give some examples of input and the various outputs that SAM produces. This will give the reader some feeling for the kinds of heavily script-based stories SAM can handle and the understanding tasks it performs.

We present three different types of story input. The first story makes references to a common script, the restaurant script, then stops the script in midstream:

Story I
John went to a restaurant. He sat down. He got mad. He left.

A 'long paraphrase' output generated by SAM for this story is as follows:

Long Paraphrase
John was hungry. He decided to go to a restaurant. He went to one. He sat down in a chair. A waiter did not go to the table. John became upset. He decided he was going to leave the restaurant. He left it.

Next is a typical boring restaurant story conforming closely to expected scriptal data:

Story II
John went to a restaurant. The hostess seated John. The hostess gave John a menu. John ordered a lobster. He was served quickly. He left a large tip. He left the restaurant.

SAM has generated summary and question-answering output for this story:

Summary

John went to a restaurant and he ate a lobster.

Question-Answering

Q: Why did the hostess give John a menu?
A: So John could ask the waiter for a meal.

Q: Why did John go to a restaurant?
A: So John could eat a meal.

Q: Did the waiter give John a menu?
A: John got the menu from the hostess.

Q: Why did John leave the waiter a large tip?
A: Probably John remembered the waiter served a lobster to him quickly.

Q: Did John pay the bill?
A: John paid the check.

The system has also provided long-paraphrase outputs for this story, not shown here, in both English and Chinese.

The final example invokes several scripts, and calls up an unusual path in one script because of an odd occurrence in an earlier one:

Story III
John went to New York by bus. On the bus he talked to an old lady. When he left the bus, he thanked the driver. He took the subway to Leone's. On the subway his pocket was picked. He got off the train and entered Leone's. He had some lasagna. When the check came, he discovered he couldn't pay. The management told him he would have to wash dishes.

Long Paraphrase

John went to a bus stop. He waited at it a few minutes. He entered a bus. The driver got the ticket from John. He went to a seat. He sat down in it. While John was on the bus an old lady and John talked. The driver took John to New York. He went to the driver. While getting off the bus John thanked the driver. John got off it.

He entered a station. He put a token in the turnstile. He went to the platform. He waited at it a few minutes. He entered a subway car. A thief went to John. The thief picked John's pocket. He went. John went to the seat. He sat down in it. The driver took John to Leone's. He left the subway car. He left the station.

He entered Leone's. He looked around inside it. He saw he could go to a table. He went to it. He sat down in the seat. He ordered some lasagna. The waiter indicated to the chef John would like him to prepare something. The chef prepared the lasagna. The waiter got it from the chef. The waiter went to the table. He served the lasagna to John. He ate it. He became full.

He asked the waiter for the check. John got it from the waiter. John read the check. John discovered he was unable to pay the check. He indicated to the waiter he was unable to pay the check. The management told John he would have to wash dishes. He entered the kitchen. He washed dishes. He left Leone's.

[paragraphing has been added to the computer output for ease of reading]

In these example stories, SAM analyzes each input sentence into a Conceptual Dependency (CD) representation. If this representation fits a script, that script is called into memory and successive inputs are matched in the script and linked up by a SAM program called the script applier. The script applier output is processed by other SAM programs depending on the type of final output desired, and English or, for Story II, Chinese is generated. The point to be stressed is that all the

'understanding' processing is done on a single data structure, the story representation constructed by the script applier. We discuss in particular the scriptal data base, the script applier and the story representation in succeeding sections. Additional details on the other parts of SAM can be found in [Schank et al, 1975].

2.Ø Situational Scripts

As implemented in SAM, a situational script is a network of CD patterns describing the major paths and turning points commonly understood by middle-class Americans to occur in stereotyped activities such as going to theaters, restaurants and supermarkets. The script idea is very similar to the independently developed 'frame system' for story understanding described in [Charniak, 1975], which is itself based loosely on the 'frame' concept [Minsky, 1974] currently used in vision research.

The patterns provided in scripts are of two general kinds: events, which we will construe broadly as including states and state-changes (2) as well as mental and physical ACTs; and causal relations among these events [Schank, 1973 and 1974].

---

(2) Certain actions like driving a car or preparing food involve complex, learned sensory-motor skills as well as scriptal knowledge. Such actions are summarized within a script as a causal relation terminating in the chief state-change effected by the action. For example, the sentence "The cook prepared the meal" is represented in LISP CD format as:

```
((CON ((ACTOR (*COOK*) <=> (*DO*)))
      LEADTO
 ((ACTOR (*MEAL*) LEAVING (*COOKSTATE* VAL (Ø))))))
```

Patterns are used in scripts not only because of the variety of possible fillers for the roles in scripts, but also to constrain the amount of information needed to identify a story input. Thus, for example, the script provides a LISP CD template like:

```
((ACTOR (X) <=> (*PTRANS*) OBJECT (X) TO (*INSIDE*
PART (RESTAURANT))))
```

to identify inputs like:

John went into Leone's.
John walked into Leone's.
John came into Leone's from the subway.

(X and RESTAURANT are dummy variables). This allows the script applier to ignore inessential features of an input (like the Instrument of the underlying ACT or the place John came from in the examples given above), and thus provides a crude beginning for a theory of forgetting.

In the present implementation, SAM possesses three 'regular' scripts, for riding a bus, for riding a subway, and for going to a restaurant (3). These scripts have been simplified in various ways. For example, all of them assume that there is only a single main actor. The bus script has been restricted to a single track' for a long-distance bus ride, and the restaurant script does not have a 'McDonald's' or a 'Le Pavillon' track. This was done primarily to have a data base capable of handling specific stories of interest available in a reasonable time, secondarily to limit the storage needed (4). Nevertheless, as

---

(3) The data base also contains script-like structures for 'weird' or 'unusual' happenings like the main actor's becoming ill, or, as in Story III, having his pocket picked. Such activities could be handled by a generalized inferencing program like the one described in [Rieger, 1975].

the examples of Section 1.1 indicate, the current scripts are a reasonable first pass at the dual problems of creating and managing this type of data structure.

2.1 Goals, Predictions and Roles in Scripts

Each situational script supplies a default goal statement which is assumed, in the absence of input from higher level cognitive processes like 'planning' [Schank and Abelson, 1975], to be what a story referring to a script is about. The restaurant script for example, defines the INGEST and the resulting state-change in hunger as the central events of a story about eating in restaurants. Closely related to the goal statement is the sequence of mutual obligations that many scripts seem to entail. Invoking the bus script, for example, implies the contract between the rider and the bus management of a PTRANS to the desired location in return for the ATRANS of the fare. Such obligations have a powerful influence on the predictions the system makes about new input. In the restaurant context, for example, an input referring to an event beyond ordering or eating is not initially expected, because these events form the initial statement of obligation. Thus the system takes longer to identify a story sequence like:

John went to a diner. He left a large tip.

Once an input about ordering has been processed, SAM is prepared

---

(4) The text for the restaurant script, presently the largest of the scripts, occupies roughly 100 blocks of PDP-10 disk storage, or about 64,000 ASCII characters.

to hear about the preparation and serving of food, actions associated with eating, or paying the bill, but not about leaving the restaurant. This is because the main actor has not fulfilled the other half of the obligation.

The binding of nominals in the story input to appropriate fillers in the script templates is accomplished in SAM by means of script variables with associated features. In the rather crude system of features presently used, each script variable is assigned a superset membership class: e. g., a hamburger is a 'food', while a waiter is a 'human'. certain variables are also given roles: e. g., a hostess or a waiter can fill the 'maitre'd' role. The former property would enable the system to distinguish between "The waiter brought Mary a hamburger" and "The waiter brought Mary the check". The latter property identifies important roles in script contexts, primarily those to which it is possible to make definite reference without previous introduction, like 'the driver', 'the cook' or 'the check'. For stories in which certain script variables are not bound, the system provides a set of default bindings for the roles not mentioned: thus, SAM fills in 'meal' for a story in which the food ordered is not explicitly named. Variables without distinguished roles default to an indefinite filler, like 'someone' for the main actor.

2.2 Script Structure

Each SAM script is organized in a top-down manner as follows: into tracks, consisting of scenes, which are in turn

composed of subscenes. Each track of a script corresponds to a manifestation of the situation differing in minor features of the script roles, or in a different ordering of the scenes. So, for example, eating in an expensive restaurant and in McDonald's share recognizable seating, ordering, paying, etc., activities, but contrast in the price of the food, type of food served, number of restaurant personnel, sequence of ordering and seating, and the like. Script scenes are organized around the main top-level acts, occurring in some definite sequence, that characterize a scriptal situation. The giving of presents, for example, would be a scene focus in a birthday party script, but putting on a party hat would not be. The latter would correspond to a subscene, perhaps within the 'preparing-to-celebrate' scene of that script. In general, subscenes are organized around acts more or less closely related to the main act of the scene, either contributing a precondition for the main act, as walking to a table precedes sitting down; or resulting from the main act, as arriving at the desired location follows from the driver's act of driving the bus. An intuitive way of identifying scene foci and scene boundaries is to visualize a script network of interwoven paths. In such a network, the scene foci would correspond to points of maximum constriction; scene boundaries to points of most constriction between foci. This essentially means that all paths through a scene go through the main act (except abort paths, discussed below), and relatively few events are at scene edges.

It is necessary, therefore, to distinguish certain events in a script: scripts, their tracks, scenes and subscenes all have 'main', 'initial' and 'final' events. For example, the main event of the 'ordering' event in a restaurant is the ordering act itself; an initial event is reading the menu; and a final event is the waiter telling the cook the order. Additionally, scripts and tracks have associated 'summaries', which refer to a script in general terms. Consider, for example, the following sentence from Story III: "John went to New York by bus". This sentence is marked in the underlying meaning representation by the SAM analyzer as a summary because of the presence of:

((ACTOR (*JOHN*) <=> (*SDO*) OBJECT ($BUS)))

in the Instrument slot (5). Such sentences have two common functions in simple stories. They may indicate that a script was invoked and completed, and no further input should be expected for this instance of the script. This function of the summary often occurs with scripts (like those associated with travelling) which tend to be used as 'instruments' of other scripts (as in getting to a restaurant or store). Alternatively, they may signal that a wider range of possible next inputs is to be expected than would be predicted if the script were entered via an initial event. For example, the story sequence initiated with a summary:

> John took a train to New York. While leaving the train, he tipped the conductor.

---

(5) The primitive ACT SDO is an extension of the primitive dummy CD ACT DO, and stands for an actor performing his script for a given situation, in this case the bus script ($BUS).

sounds more natural than a sequence beginning with an initial event:

> John got on a train. While leaving the train, he tipped the conductor.

These two functions of the summary contrast widely in the range of predictions they invoke. However, additional inputs after a summary, as in the example above, often give the psychological feeling of 'afterthoughts'.

Scenes are built up out of subscenes, which usually contain a single chunk of causal chain or 'path'. In SAM scripts, these paths are assigned a 'value' to indicate roughly their normality in the scriptal context. Several pathvalues have been found useful in setting up the story representation. At one end of the normality range is 'default', which designates the path the script applier takes through a scene when the input does not explicitly refer to it. For example, the input sequence:

> John went to Consiglio's. He ordered lasagna.

makes no mention of John's sitting down, which would commonly be assumed in this situation. The system, following the default path, would fill in that John probably looked around inside the restaurant, saw an empty table, walked over to it, etc. Next on the normality scale is 'nominal', designating paths which are usual in the script, not involving errors or obstructions in the normal flow of events. The sentences in Story II which refer to the hostess are examples of nominal inputs. Finally, there are the 'interference/resolution' paths in a script. These are followed when an event occurs which blocks the normal functioning

of the script. In a restaurant, for example, having to wait for a table is a mild interference; its resolution occurs when one becomes available. More serious because it conflicts directly with the goal/obligation structure of the script is the main actor's discovery that he has no money to pay the bill. This is resolved in Story III by his doing dishes. An extreme example of an interference is the main actor's becoming irritated when a waiter fails to take his order, as in Story I, followed by his leaving the restaurant. When this happens, the script is said to have taken an 'abort' path.

In addition to the above, certain incomplete paths, i. e., paths having no direct consequences within the script, have been included in the scriptal data base. The most important of these incomplete paths are the inferences from, and preconditions for, the events in the direct causal paths. Lumped under the pathvalue 'inference', these subsidiary events identify crucial resultative and enabling links which are useful in particular for question-answering [Lehnert, 1975]. For example, the main path event 'John entered the train' has attached the precondition that the train must have arrived at the platform, which in turn is given as a result of the driver's bringing the train to the station. Similarly, a result of the main path event 'John paid the bill' is that he has less money than previously. Both of these types of path amount to a selection among the vast number of inferences that could be made from the main path event by an inferencing mechanism like Rieger's Conceptual Memory program [Rieger, 1975].

A special class of resultative inferences are those common events which are potentialized by main path events, though they may not occur in a given story. Labelled with the pathvalue 'parallel', these events may either occur often in a specific context without having important consequences, as in "The waiter filled John's water glass"; or they may happen in almost any context without contributing much to the story, as in the sentence "On the bus, John talked to an old lady", from Story III. Since such parallel paths often lead nowhere, they are good candidates for being forgotten.

3.0 The Script Applier

Construction of a story representation from CD input supplied by the SAM analyzer is the job of the script applier (6). Under control of the SAM executive, the applier locates each new input in its collection of situational scripts, links it up with what has gone before, and makes predictions about what is likely to happen next. Since the SAM system as a whole is intended to model human understanding of simple, script-like stories, the script applier organizes its output into a form suitable for subsequent summary, paraphrase and question-answering activities.

In the course of fitting a new input into the story

---

(6) The current version of the applier is programmed in MLISP/LISP 1.6 and runs in an 85K core image on a PDP-10 computer. Processing of Story III, the longest story attempted to date, took approximately 8 minutes with SAM as the single user of the timesharing system.

representation, the applier performs several important subtasks. Identifying an input often requires an implicit job of reference specification. For example, in the sentence from Story III beginning "When the check came...", there is surface ambiguity, reflected in the parser's output, regarding donor and recipient. This ambiguity is settled in the restaurant context by the assumption that the recipient is the main actor and that the donor is a member of the restaurant staff, preferably the waiter. An allied problem arises when the applier, in placing a new conceptualization in the story representation, determines the relevant time relations. Certain types of time data are computed from the output conceptualization itself: for example, the relation between an MTRANS and its MOBJECT, which may determine whether 'remember' or 'ask for' is appropriate in the final output. Other time relations are defined by the causal structure of the script itself: thus 'eating' follows 'ordering'.

More complex time-order computations have to be made when the applier identifies two or more 'simple' conceptualizations in a compound input derived from sentences containing ambiguous words like 'during' or 'when'. Examples of this were encountered during the processing of Story III, for example, in the sentence 'When he left the bus, he thanked the driver'. The system resolves this compound input into the plausible sequence of a PTRANS to the driver, the MTRANS of the 'thanking , and the PTRANS off the bus.

3.1 Story Representation

The output of the script applier consists of linked story segments, one per script invoked, giving the particular script paths traversed by the input story. The backbone of the story representation is the eventlist of all the acts and state-changes that took place. The eventlist is doubly linked, causally and temporally, with the type of causation and time relations filled in within a story segment by the applier.

Attached to the eventlist are the appropriate, instantiated preconditions, inferences and parallel events for each main path event. As discussed above, the inferences and preconditions have been selected for their expected utility in question-answering.

Each story segment is identified by a label which gives access to important properties of the segment: what script it came from; what the particulars were of the script summary, maincon, entrycon, and exitcon this time through; and what interference/resolution cycles were encountered. Additionally, pointers are provided to extra-scriptal 'weird' events that happened in the story. At the top, the global identifier STORY gives the gross structure of the story in terms of sequential, parallel and nested scripts and the weird things. This hierarchical organization facilitates summary and short paraphrase processing, while retaining the fine structure needed for extended paraphrasing and question-answering.

Story III illustrates most of the present capabilities of the SAM script applier in story understanding. The applier accepts a CD representation of the nine sentences in turn from the analyzer and builds an eventlist consisting of 56 main path conceptualizations and 39 associated preconditions/inferences. The 'parallel' events of John talking to the old lady and the bus driver also appear in the eventlist. The eventlist is divided into four story segments, one each for the bus, subway and restaurant scripts and one for the 'weird' robbery event. The identifier for the subway segment is marked as containing the weird event, as is the global STORY. The restaurant segment contains the interference/resolution pair 'unable to pay/wash dishes'. Additionally, the lack of money encountered during the paying scene was checked with the SAM executive during the processing of Story III, since it violates one of the prime preconditions of the restaurant script. Since the executive found that the loss of money was a consequence of the stealing event that occurred earlier, this event is not marked as weird. Appropriate summaries are provided for each story segment. At the top, STORY contains the information that the four segments are organized as a sequence of bus, subway and restaurant, with the pickpocket event nested inside the subway segment.

4.0 Future Work

As the examples show, SAM is capable of handling fairly complex stories in its present state of development. However, several extensions and additions to the scriptal data base and the script applier appear to be needed before SAM can achieve its

ultimate potential.

First, a more flexible method of pattern-matching is required so that the full diversity of input role-fillers can be accommodated. A method of comparing features of nominals in the parser output to the appropriate script variables is needed so that over- or underspecified inputs can be correctly identified. For example, the applier should be able to recognize the phrase 'the restaurant' as a partially specified instance of 'Leone's', found earlier.

As an extension of this, input conceptualizations of a descriptive nature (e. g., "The restaurant was of red brick") need to be processed in a way that allows the system to update its 'image' of the role-fillers in a script. The facilities needed are similar to those provided by the 'occurrence set' in Rieger's Conceptual Memory program [Rieger, 1975].

The most important problem to be faced, however, is the generalization of the story representation to handle stories with several main actors, or with non-synchronous events. It is clear that the simple linear eventlist structure described in Section 3.1 would not be adequate for even such a simple story sequence as:

> "The cook made the lasagna. Meanwhile the wine steward poured the wine."

4.1 Acknowledgement

The programs discussed here are only a part of the SAM system, and a great deal of credit is due to my co-workers in the

Yale AI Project: to Professors Roger Schank and Bob Abelson for the theory on which SAM is based and for their overall guidance; to Dr. Chris Riesbeck for valuable discussion and criticism, as well as a substantial part of the programming effort; and to Gerry DeJong, Leila Habib, Wendy Lehnert, Jim Meehan, Dick Proudfoot, Wally Stutzman and Bob Wilensky.

References

Schank and Abelson 1975
R. C. Schank and R. P. Abelson, "Scripts, Plans and Knowledge", Proceedings of the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, USSR, 1975.

Schank 1973
R. C. Schank, "Causality and Reasoning", Technical Report No. 1, Instituto per gli studi semantici e cognitivi, Castagnola, Switzerland, 1973.

Schank 1974
R. C. Schank, "Understanding Paragraphs", Technical Report No. 6, Instituto per gli studi semantici e cognitivi, Castagnola, Switzerland, 1974.

Schank et al 1975
R. C. Schank and the Yale AI Project, "SAM--A Story Understander", Research Report No. 43, Yale University Department of Computer Science, 1975

Lehnert 1975
W. P. Lehnert, "What makes SAM run? Script-Based Techniques for Question Answering", Proceedings of the Conference on Theoretical Issues in Natural Language Processing, edited by R. Schank and B. Nash-Webber, 1975.

Charniak 1975
E. Charniak, "Organization and Inference in a Frame-Like System of Common Sense Knowledge", Proceedings of the Conference on Theoretical Issues in Natural Language Processing, edited by R. Schank and B. Nash-Webber, 1975.

Minsky 1974
M. Minsky, "Frame-Systems", MIT AI Memorandum, 1974.

Rieger 1975
C. Rieger, "Conceptual Memory", in Conceptual Information Processing, R. Schank (ed.), North Holland, 1975.

American Journal of Computational Linguistics *Microfiche 35 : 70*

The Conceptual Description of Physical Activities

Norman Badler

Department of Computer and Information Science
The Moore School of Electrical Engineering
University of Pennsylvania
Philadelphia 19174

ABSTRACT

A system has been designed to translate connected sequences of visual images of physical activities into conceptual descriptions. The representation of such activities is based on a canonical verb of motion so that the conceptual description will be compatible with semantic networks in natural language understanding systems. A case structure is described which is derived from the kinds of information obtainable in image data. A possible solution is presented to the problem of segmenting the temporal information stream into linguistically and physically meaningful events. An example is given for a simple scenario, showing part of the derivation of the lowest level events. The results of applying certain condensations to these events show how details can be systematically eliminated to produce simpler, more general, and hence shorter, descriptions.

This research was primarily supported by Canadian Defense Research Board grant 9820-11, and partially by National Science Foundation grant ENG75-10535.

If we view a motion picture such as illustrated in Figure 1, we are able to give a description of the physical activities in the scenario. This description is linguistic in the sense that the words used express our recognition of objects and movements as conceptual entities. A system for performing a sizeable part of this transformation of visual data into conceptual descriptions has been designed. It is described in Badler (1975); here we will present one small part of the system which is concerned with the organization of abstracted data from successive images of the scenario.

We are interested in a possible solution to the following problem: Given that a conceptual description of a scenario is to be generated, how is it decided where one verb instance starts and another ends? In other words, we seek computational criteria which separate visual experience into discrete "chunks" or events. By organizing the representation of an event into a case structure for a canonical motion verb, events can be described in linguistic terms. Verbs of motion have been investigated directly or indirectly by Miller (1972), Hendrix et al (1973a, 1973b), Martin (1973), and Schank (1973); semantic databases using variants of case structure verb representations (Fillmore(1968)) include Winograd (1972), Rumelhart et al (1972), and Simmons (1973).

We are concerned with physical movements of rigid or jointed objects so that motions may be restricted to translations and rotations. Objects may appear or disappear and the observer is free to move about. The resulting activities are combinations of these where observer motions are factored out if at all possible. We assume that the scenarios contain recognizable objects exhibiting physically possible, and preferably natural, motions.

A particular activity might consist of a single event, a sequence of events, sets of event sequences, or hierarchic organizations of events. The concept of "walking" is a good example of the last. Events are the basic building blocks of the conceptual description, and our events indicate the motions of objects. The interpretation of motion in terms of causal relationships is generally

Figure 1. The moving car scenario

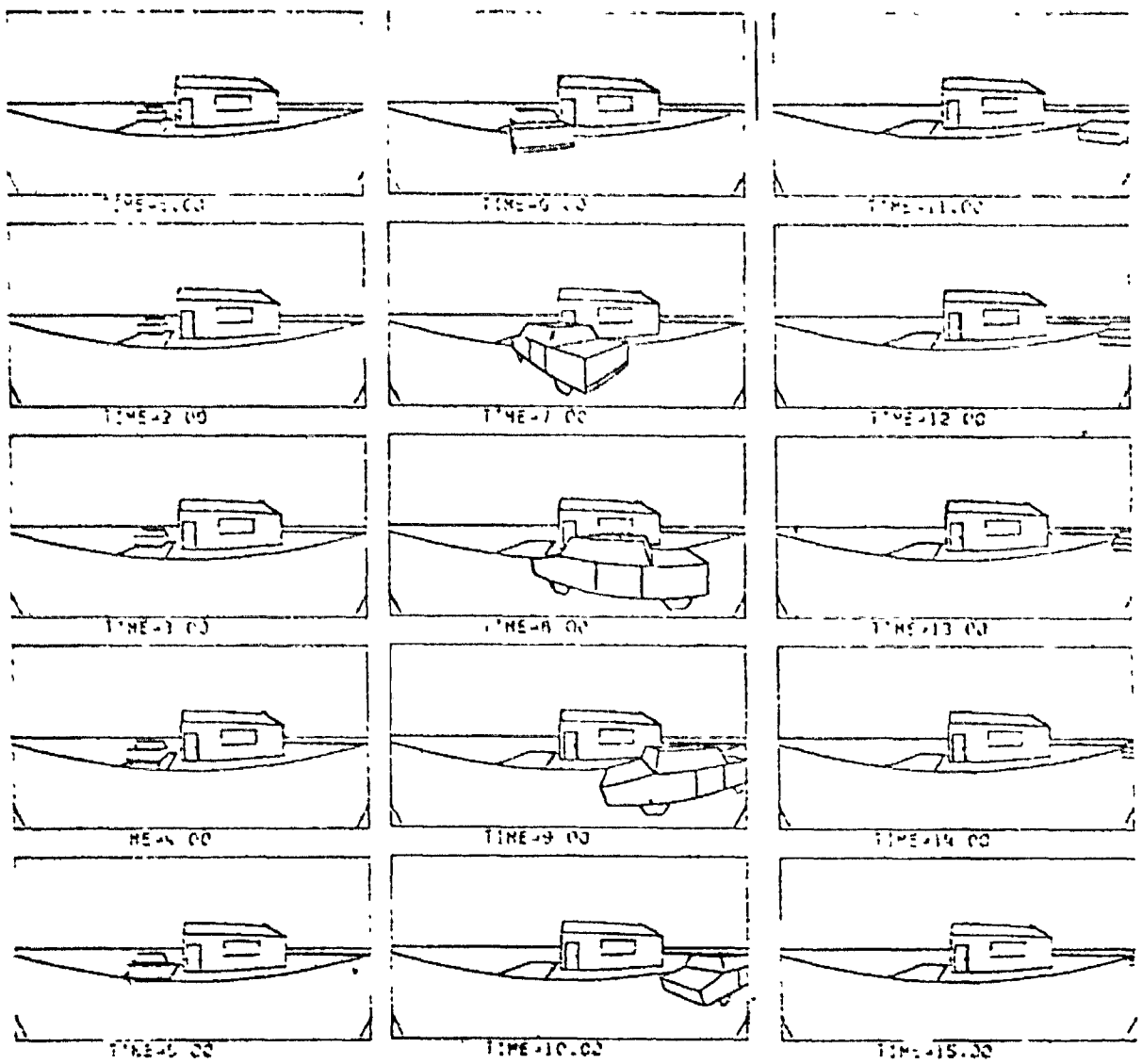


Table 1

Adverbials

Type	Relationships	Set of Concepts
1	between the orientation and trajectory or axis of an object	BACKWARD, FORWARD, SIDEWAYS AROUND, OVER,CLOCKWISE, COUNTERCLOCKWISE
2	between the trajectory of an object and fixed world directions -	DOWN(WARD),UP(WARD),NORTHWARD SOUTHWARD,EASTWARD,WESTWARD
3	changing between objects.	ACROSS,AGAINST,ALONG,APART, AROUND,AWAY,AWAY-FROM, BEHIND,BY,FROM,IN,INTO,OFF, OFF-OF,ON,ONTO,OUT,OUT-OF, OVER,THROUGH,TO,TOGETHER, UNDER
4	indicative of source and target	AWAY-FROM,IN-THE-DIRECTION-OF, IN(WARD),OUT(WARD),TOWARD
5	between the path of an object and other (moving) objects	AFTER, AHEAD-OF,ALONG,APART TOGETHER,WITH
6	between an event and a previous event	BACK-AND-FORTH,TO-AND-FRO, UP-AND-DOWN,BACK,THROUGH

beyond the scope of the current system, although a semantic inference component could be included. Our descriptions consist mostly of observation of motion in context rather than explanation of why motion occurred.

The general descriptive methodology is to keep only one static relational description of the scenario, that of the current image. Changes between it and the next sequential image are described by storing the names of changes in event nodes in a semantic network. In general, names of changes correspond to adverbs or prepositions (adverbials) describing directions or changing static relationships. Computational definitions for the set of adverbials in Table 1 appear in Badler (1975). We are only concerned with the senses of the adverbials pertaining to movement. Definitions are implemented as demons: procedures which are activated, the executed, by the successive appearance of certain assertions in the image description or current conceptual database. These demons are related to those of Charniak (1972), although our use of them, their numbers, and their organization are simplified and restricted. They are used to recognize or classify properties or changes and to generate the hierarchic descriptive structure. An essential feature of this methodology is that the descriptions are continually condensed by this change abstraction process; descriptions grow in depth rather than length.

The semantic information stored for each object in the scenario includes its TYPE, structural SUB-PARTs, VISIBILITY, MOBILITY, LOCATION ORIENTATION, and SIZE. Most of these properties are determined from the image sequence, but some are stored in object models (indexed by TYPE) in the semantic network.

The events are also nodes in the semantic network. Each object is potentially the SUBJECT of an event node. A sequence of event nodes forms a history of movement of an object; only the latest node in the sequence is active. The set of active event nodes describes the current events in the scenario seen so far. The cases of the event node along with their approximate definitions follow.

SUBJECT: An object which is exhibiting movement.
AGENT: A motile object which contacts the SUBJECT.
INSTRUMENT: A moving object which contacts the SUBJECT.
REFERENCE: A pair of object features (on a fixed object) which are used to fix absolute directions independent of the observer's position.
DIRECTION: A temporally-ordered list of adverbials and their associated objects which apply to this SUBJECT.
TRAJECTORY: The spatial direction of a location change of the SUBJECT.
VELOCITY: The approximate magnitude of the velocity of the SUBJECT along the TRAJECTORY; it includes a RATES list containing STARTS, STOPS and (optionally) INCREASES or DECREASES.
AXIS: The spatial direction of an axis of an orientation change (rotation) of the SUBJECT.
ANGULAR-VELOCITY: Similar to VELOCITY, except for rotation about the AXIS.
NEXT: The temporal successor event node having the same SUBJECT.
START-TIME: The time of the onset of the event.
END-TIME: The time of the termination of the event.
REPEAT-PATH: A list of event nodes which form a repeating sequence.

These cases differ from Miller's (1972) primarily in the lack of a "permissive" case and our separation of the TRAJECTORY and AXIS cases. REFERENCE is new; one of its uses is to resolve descriptions of the same event from different viewpoints. The explicit times could be replaced by temporal relations. Miller's reflexive/objective distinction is not needed as each moving object has its own event nodes, regardless of the AGENT.

A few necessary definitions follow before the presentation of the event generation algorithm.

A null event node has all its cases NIL or zero except START-TIME, END-TIME, and perhaps NEXT.

An event node is terminated when it has a non-NIL NEXT value.

The function CREATE-EVENT-NODE (property pairs) creates an event node with the indicated case values, returning the node as a result.

To compare successive values of numerical properties , a queue is associated with the case in current event nodes only. The front of the queue is represented by "*": the place where new information is stored. The queues have length three; the three positions will be referenced by prefixing

the case name with either "NEW", "CURRENT", or "LAST". A function SHIFT manipulates property queues when they require updating:

```
LAST-property: = CURRENT-property;
CURRENT-property: = NEW-property;
NEW-property: = *
```

The time will be abbreviated by TN and TL. For a particular event node E:

```
TN: = NEW-END-TIME (E);
TC: = CURRENT-END-TIME (E);
```

Thus TN is always equal to the present image time.

Now we can present the algorithm for the demon which controls the construction of the entire event graph. It is executed once for each image when all lower level demons have finished; it creates,terminates, or updates each current event node.

A.1. Creating event nodes.

A.1.1. An event node E is created when a mobile object first becomes visible and identifiable as an object.

```
E: = CREATE-EVENT-NODE((SUBJECT object-node)
                       (VELOCITY(* 0. 0.))
                       (ANGULAR-VELOCITY (* 0. 0.))
                       (START-TIME NIL)
                       (END-TIME (* TN TN))  ).
```

The NIL START-TIME has the interpretation that we do not know what was happening to this object prior to time TN.

A.1.2. An event node E is created when a jointed part of the parent object with current event node EP is first observed to move relative to the parent, for example, an arm relative to a person's body.

```
TC: = CURRENT-END-TIME(EP);
E: = CREATE-EVENT-NODE( (SUBJECT object-part-node)
                        (AGENT parent-object-node)
                        (INSTRUMENT joint-node)
                        (REFERENCE ...)
                        (DIRECTION  ...)
                        (TRAJECTORY ...)
                        (VELOCITY ...)
                        (AXIS ...)
                        (ANGULAR-VELOCITY ...)
                        (START-TIME TC)
                        (END-TIME (TN TC TC))  ).
```

This is interpreted as the parent object moving the part using the joint as the "instrument". Any appropriate attributes are placed in the NEW-property positions. The node E is then immediately terminated (A.1.3).

A.1.3. An event node E2 is created whenever another event node E1 is terminated.

```
TC: = CURRENT-END-TIME(E1);
NEXT(E1): = CREATE-EVENT-NODE(
                    (SUBJECT...)
                    (AGENT...)
                    (INSTRUMENT...)
                    (REFERENCE...)
                    (DIRECTION...)
                    (TRAJECTORY SHIFT(TRAJECTORY(E1)))
                    (VELOCITY SHIFT(VELOCITY(E1)))
                    (AXIS SHIFT(AXIS(E1)))
                    (ANGULAR-VELOCITY SHIFT(ANGULAR-
                              VELOCITY(E1)))
                    (START-TIME TC)
                    (END-TIME SHIFT(END-TIME(E1)));
E2: = NEXT(E1).
```

SUBJECT, AGENT, INSTRUMENT, REFERENCE, and DIRECTION are those which were present at termination of the previous node, subject to any additional conditions that changes in these may require.

A.2. Terminating event nodes. An event node E is terminated when there are significant changes in its properties. All queue structures are deleted.

```
END-TIME(E): = CURRENT-END-TIME(E);
TRAJECTORY(E): = CURRENT-TRAJECTORY(E);
AXIS(E): = CURRENT-AXIS(E);
VELOCITY(E): = (CURRENT-VELOCITY(E) RATES(VELOCITY(E)));
ANGULAR-VELOCITY(E): = (CURRENT-ANGULAR-VELOCITY(E)
                              RATES(ANGULAR-VELOCITY(E))).
```

The DIRECTION list is unaltered except that the terminating adverbial (s) may be added to DIRECTION(E) rather than to DIRECTION(NEXT(E)) (see A.2.5.).

A.2.1. Changes in SUBJECT. The assumptions of object rigidity and permanence preclude changes in an object.

A.2.2/3. Changes in AGENT and INSTRUMENT. These must be preceded by changes in CONTACT relations between objects and the SUBJECT. See A.2.5 on DIRECTION.

A.2.4. Changes in REFERENCE. A change in the REFERENCE features forces termination of every event node referencing those features, as such changes are usually caused by spatial or temporal discontinuities in the scenario.

A.2.5. Changes in DIRECTION.

Changes in type (1) adverbials must be preceded by changes in TRAJECTORY, VELOCITY, AXIS, or ANGULAR-VELOCITY, because a relationship between an orientation and a TRAJECTORY or AXIS cannot change without at least one of the four cases changing. Changes in BACKWARD, FORWARD, and SIDEWAYS cause termination; this may occur with no orientation change if the TRAJECTORY has a non-zero derivative. For example, move a box in a circle while keeping its orientation constant.

Changes in type (2) adverbials must be preceded by a change in TRAJECTORY, but some of these changes may be too slight to cause termination from the TRAJECTORY criteria. (A.2.6.). Changes from UP to DOWN or vice versa are the only ones in this group causing termination.

Changes in type (3) adverbials terminate event nodes if and only if there is a change in a CONTACT relation or a VISIBILITY property. If the CONTACT is made or the VISIBILITY established, the adverbial goes into the new node's DIRECTION list. If the CONTACT is broken or VISIBILITY lost, the adverbial remains on the front of the terminated node's DIRECTION list.

Since the type (4) adverbials are only indicators of current source and target, these do not change unless the path of the SUBJECT changes or the target object moves. Therefore no terminations arise from this group.

The type (5) adverbials relate paths of the SUBJECT to other objects. They cause termination when they come into effect, and terminate their own nodes when they cease to describe the path.

The type (6) adverbials include higher level events and the basic repetitions. These all terminate the current event node. The repeated events (for example, BACK-AND-FORTH) are terminated when the repetition appears to cease.

A.2.6. Changes in TRAJECTORY. The changes in TRAJECTORY that are most important are those which change its derivative significantly. A change in the derivative from or to zero can be used (the start or end of a turn), but only the start is actually used for termination. Once the turn is begun, how it ends is unimportant since the final (current) trajectory is always saved.

The other termination case watches for a momentarily large derivative which settles back to smaller values. This indicates a probable collision. It is of crucial importance in inferring CONTACT relations between objects when none were (or could be) directly observed.

A.2.7. Changes in VELOCITY. A change in VELOCITY from zero to a positive value (from a positive value to zero) terminates the current event node and enters STARTS (STOPS) in the new node's (old node's) VELOCITY RATES list.

A.2.8. Changes in AXIS. A reversal of rotation terminates the event node. This corresponds to a change in AXIS to the opposite direction, with no intermediate values.

A.2.9. Changes in ANGULAR-VELOCITY. A change in ANGULAR-VELOCITY from zero to a positive value (from a positive value to zero) terminate the current event node and enters STARTS (STOPS) in the new node s (old node's) ANGULAR-VELOCITY RATES list.

A.2.10. Changes in NEXT are not meaningful.

A.2.11/12. Changes in START-TIME and END-TIME are not meaningful.

A.2.13. Changes in REPEAT-PATH. When new data fails to match the appropriate sub-event node of a REPEAT-PATH event node E, E is terminated. The definition of "match" for the basic repetitions appears in Badler (1975). The problem, in general, remains open. See, for example, Becker (1973).

A.3. Maintaining event nodes. If the new assertions do not cause termination of the event node, the property queues are merely shifted:

TRAJECTORY(E): = SHIFT(TRAJECTORY(E));
VELOCITY(E): = SHIFT(VELOCITY(E));
AXIS(E): = SHIFT(AXIS(E));
ANGULAR-VELOCITY(E) : = SHIFT(ANGULAR-VELOCITY(E));
END-TIME(E): = SHIFT(END-TIME(E)).

What does an event mean? This algorithm motivates a theorem that the events generated are the finest meaningful partition of the movements in the image sequence into distinct activities. The hypothesis of the assertion is the natural environment being observed and the linguistically-based conceptual description desired. The conclusion is that an event node produced from this algorithm describes either the lack of motion or else an unimpeded, simple linear or smoothly curving (or rotating) motion of the SUBJECT with no CONTACT changes. In addition, the orientation of the SUBJECT does not change much with respect to the trajectory. The proof of this assertion follows directly from the choice of termination conditions.

We will apply this algorithm to data obtained from each of the images in Figure 1. The lower front edge of the house is arbitrarily chosen as the REFERENCE feature; NORTH is toward the right of each image. We will not discuss the computation of the static relations from each image, only list in Table 2 the changes in the static description from image-to-image. Trajectory and rotation data are omitted for simplicity, although changes of significance are indicated.

If we "write out" the event node sequence using the canonical motion verbs MOVES and TURNS with the adverbial phrases from the RATES and DIRECTION lists, we obtain the following lengthy, but accurate, description:

C.1 There is a CAR.
C.2 The CAR STARTS MOVING TOWARD the OBSERVER and EASTWARD, then ONTO the ROAD.
C.3 The CAR, while GOING FORWARD, STARTS TURNING, MOVES TOWARD the OBSERVER and EASTWARD, then NORTHWARD-AND-EASTWARD, then FROM the DRIVEWAY and OUT-OF the DRIVEWAY, then OFF-OF the DRIVEWAY.

Table 2

Selected assertions and changes involved in the description of Figure 1.

Time	Action	Static Assertion	Event Assertion	Result
1	ADD	IN-FRONT-OF(CAR OBSERVER)		create C1
	ADD	IN-BACK-OF(CAR HOUSE)		
	ADD	RIGHT-OF(CAR HOUSE)		
	ADD	NEAR-TO(CAR HOUSE)		
	ADD	SURROUNDED-BY(CAR DRIVEWAY)		
	ADD	LEFT-OF(CAR DRIVEWAY)		
	ADD	IN-BACK-OF(CAR DRIVEWAY)		
	ADD	RIGHT-OF(CAR DRIVEWAY)		
	ADD	AT(CAR DRIVEWAY)		
	ADD	SUPPORTED-BY(CAR DRIVEWAY)		
3	DELETE	IN-BACK-OF(CAR HOUSE)	VELOCITY (STARTS)	terminate C1 (A.2.7.)
			EASTWARD	--
			TOWARD OBSERVER	--
5	DELETE	IN-BACK-OF(CAR DRIVEWAY)	TRAJECTORY change	terminate C2 (A.2.6.)
	ADD	SUPPORTED-BY(CAR ROAD)	ONTO ROAD	terminate C2 (A.2.5.)
	ADD	IN-FRONT-OF(CAR DRIVEWAY)	ANGULAR-VELOCITY (STARTS)	terminate C2 (A.2.9.)
6	ADD	IN-FRONT-OF(CAR HOUSE)	NORTHWARD-AND-EASTWARD	--
7	DELETE	LEFT-OF(CAR DRIVEWAY)	OUT-OF DRIVEWAY	--
	DELETE	SURROUNDED-BY(CAR DRIVEWAY)		
	DELETE	AT(CAR DRIVEWAY)	FROM DRIVEWAY	--
	ADD	NEAR-TO(CAR DRIVEWAY)	FORWARD	--
8	DELETE	SUPPORTED-BY(CAR DRIVEWAY)	OFF-OF DRIVEWAY	terminate C3 (A.2.5.)
9			NORTHWARD	--
10	DELETE	NEAR-TO(CAR DRIVEWAY)	AROUND HOUSE	--
	ADD	LEFT-OF(CAR HOUSE)	AWAY-FROM DRIVEWAY	--
	ADD	FAR-FROM(CAR DRIVEWAY)		
12	DELETE	NEAR-TO(CAR HOUSE)	AWAY-FROM HOUSE	--
	ADD	FAR-FROM(CAR HOUSE)		
			ANGULAR-VELOCITY (STOPS)	terminate C4 (A.2.9.)
15	DELETE	VISIBILITY(CAR VISIBLE)	AWAY	terminate C5 (A.2.5.)

Notes: Relations with HOUSE use the house front orientation, not the observer's front.

Termination of Ci creates Ci+1 by A.1.3.

C.4 The CAR, while GOING FORWARD, MOVES NORTHWARD-AND-EASTWARD, then NORTHWARD, then AROUND the HOUSE and AWAY-FROM the DRIVEWAY, then AWAY-FROM the HOUSE and STOPS TURNING.

C.5 The CAR, while GOING FORWARD, MOVES NORTHWARD, then AWAY.

The canonical form follows easily from the case representation and the DIRECTION list orderings. The directional adverbials FORWARD, BACKWARD and SIDEWAYS are interpreted as lasting the duration of the event, hence are written as "while GOING..." clauses. STARTS is always interpreted at the beginning of the sentence, STOPS at the end. The termination conditions assure its correctness.

There is much redundancy in this description, but it is only the lowest level, after all, and many activities span several events. Two sets of condensations are applied by demons that watch over terminated event nodes. The first set is mostly concerned with interpreting certain null events caused by the image sampling rate and removing trajectory changes which prove to be insignificant. The second set of demons removes adverbials referring to directions in the support plane, removes RATES terms except STOPS, and generalizes redundant adverbials referring to the same object. The result of applying these condensations is:

C.2 The CAR MOVES TOWARD the OBSERVER, then ONTO the ROAD.

C.3 The CAR, while GOING FORWARD, MOVES TOWARD the OBSERVER, then FROM the DRIVEWAY.

C.4 The CAR, while GOING FORWARD, MOVES AROUND the HOUSE and AWAY-FROM the DRIVEWAY, then AWAY-FROM the HOUSE, then STOPS TURNING.

C.5 The CAR, while GOING FORWARD, MOVES AWAY.

Another condensation can be applied for the sake of less redundant output. It does not, however, permanently affect the database:

The CAR MOVES TOWARD the OBSERVER, then ONTO the ROAD, while GOING FORWARD, then FROM the DRIVEWAY, then AROUND the HOUSE, then AWAY-FROM the HOUSE, then STOPS TURNING, then MOVES AWAY.

Note that FROM the DRIVEWAY follows ONTO the ROAD. This is due to the pictorial configuration: the car is on the road before it leaves the driveway. The position of the "while GOING FORWARD" phrase could be shifted backwards in time to the beginning of the translatory motion, but this may be risky in general. We will leave it where it is, since this is primarily a higher level linguistic matter.

By applying demons which recognize instances of specific motion verbs to the individual event nodes, then condensing as above, we get:

> The CAR APPROACHES, then MOVES ONTO the ROAD, then LEAVES the DRIVEWAY, then TURNS AROUND the HOUSE, then DRIVES AWAY-FROM the HOUSE, then STOPS TURNING, then DRIVES AWAY.

The major awkwardness with this last description is that it relates the car to every other object in the scene. Normally one object or another would be the focus of attention and statements would be made regarding its role. Such manipulations of the descriptions are yet unclear.

In conclusion, we have outlined a small part of a system designed to translate sequences of images into linguistic semantic structures. Space permitted us only one example, but the method also yields descriptions for scenarios containing observer movement and jointed objects (such as walking persons). The availability of low level data has significantly shaped the definitions of the adverbials and motion verbs. Further work on these definitions, especially motion verbs, is anticipated. We expect that the integration of vision and language systems will benefit both domains by sharing in the specification of representational structures and description processes.

References

Badler, N. (1975). "Temporal scene analysis: Conceptual descriptions of object movements." University of Toronto, Department of Computer Science, Technical Report No. 80, February 1975.

Becker, J. (1973). "A model for the encoding of experiential information." In Computer Models of Thought and Language, Schank, R. and Colby, K. (eds.), W.H. Freeman & Co., San Francisco, 1973, pp. 396-434.

Charniak, E. (1972). "Toward a model of children's story comprehension." MIT Artificial Intelligence Report TR-266, December 1972.

Fillmore, C. (1968). "The case for case." In Universals in Linguistic Theory, Bach, E. and Harms, R. (eds.), Holt, Rinehart, and Winston, Inc., Chicago, 1968.

Hendrix, G. (1973a.). "Modeling simultaneous actions and continuous processes." Artificial Intelligence 4, Winter 1973, pp. 145-180.

Hendrix, G., Thompson, C. and Slocum, J. (1973b). "Language processing via canonical verbs and semantic models." Third International Joint Conference on Artificial Intelligence, August 1973, pp. 262-269.

Martin, W. (1973). "The things that really matter - A Theory of prepositions, semantic cases, and semantic type checking." Automatic Programming Group, Internal Memo 13, MIT Project MAC, 1973.

Miller, G. (1972). "English verbs of motion: A case study in semantics and lexical memory." In Coding Processes and Human Memory, Melton, A. and Martin, E. (eds.), V.H. Winston & Sons, Washington, D.C., 1973, pp. 335-372.

Rumelhart, D., Lindsay, P. and Norman D. (1972). "A process model for long term memory." In Organization of Memory, Tulving, E. and Donaldson, W. (eds.), Academic Press, New York, 1972, pp. 197-246.

Schank, R. (1973). "The fourteen primitive actions and their inferences." Stanford A.I. Laboratory Memo AIM-183, 1973.

Simmons, R. (1973). "Semantic networks: Their computation and use in understanding English sentences." In Computer Models of Thought and Language, Schank, R. and Colby, K. (eds.), W.H. Freeman & Co., San Francisco, 1973, pp. 63-113.

Winograd, T. (1972). Understanding Natural Language, Academic Press, New York, 1972.

American Journal of Computational Linguistics *Microfiche 35 : 84*

A Frame Analysis of American Sign Language

Judy Anne Kegl and Nancy Chinchor

Department of Linguistics
University of Massachusetts
Amherst 01002

ABSTRACT

This paper is a justification for the use of frame analysis as a linguistic theory of American Sign Language. We give examples to illustrate how frame analysis captures many of the important features of ASL.

0. Introduction

From a linguistic standpoint, we are interested in language processing systems for the claims that they make about language in general. Our interests in those claims leads us to examine what implications they may have for the analysis of languages other than English. The data from American Sign Language (ASL) is important because it is indicative of the way people perceive and represent events. This linguistic data requires careful analysis and much psychological insight before it can be used as evidence for any particular theory of representation of visual knowledge of events. We have tried to bring together some ideas from artificial intelligence, linguistics, and psycholinguistics in order to analyze the data from ASL.

The major framework we have adopted from AI is that of frames. Minsky's introduction of frames as a way of representing knowledge and the further

formulations of frames and related notions by Winograd and Fillmore form the bases for our frame analysis. We rely heavily on the work done by psycholinguists on visual perception as a justification for using frame analysis. Further justification comes as a result of the work of linguists and psycholinguists on ASL and the visual perception of the deaf.

The two most direct sources for our analysis of ASL are Reid (1974) and Thompson (1975). Reid's paper presents a clear and useful distinction between the linguistic level of the sentence and the conceptual level of the image. The sentence is a generalization and the image is an instantiation of that generalization. However, "the units in a sentence are not just realized as 'parts' of a whole represented in the image by the individual participants, rather these units act reciprocally to determine jointly the character of the related participants and to unite them into a system of dependencies." At the level of the sentence the verb is all-important because it governs the relations that exist between the nouns. However, it has no direct representation in the image; it is merely embodied in the structure of the image. Thompson's paper gives guidelines for using frames in linguistic analysis. His definitions of key concepts and his examples of frames for English have been a model for our analysis.

1. American Sign Language

ASL is the language of many deaf people in the US. There is a continuum encompassing the many version of several sign systems. ASL is a manual language composed of signs, fingerspelling, and occasional initialization of signs. It is in no way a signed version of English but is rather an independent language as different from English as is French or Japanese.

ASL is a visual language. This visual modality allows it not only a temporal but also a multidimensional spatial framework as well as freedom from many of the constraints normally put on a linear language. Many spatial relations can be preserved in miniature in what has been referred to in the sign literature as a visual analog. For example, the sentence, 'Fred stood in front of Harry,' does not necessitate a linear description. It can be represented by the indexicalized marker for FRED being positioned in the signing space in front of the one for HARRY. It is with respect to the specification of location and the use of deictic elements that sign most clearly distinguishes itself from spoken languages. This and other related problems in sign will be examined later in this paper. Focusing on the aspects of visual analog and deixis does not imply that sign does not employ many of the linear and temporal devices used in spoken languages, but rather that these devices serve different functions.

ASL is linearly ordered with respect to a standard method for presenting a scenario. The order of presentation is usually ground, then figures, then the action or relation involved. A room would be specified, then a door, then relevant furniture, then participants in an action. Generally,signs are presented in such a way as to allow further reference to them even if this referencing was not intended when the element was introduced into the discourse.

A relational grammar (Perlmutter and Postal) can be useful in describing ASL. Their grammar focuses on the relations of various participants in an action to the verb. The notion of subject can be related to what Friedman calls the Agent (AGENT-PATIENT) or what Reid calls the causer (CAUSER-AFFECTED ELEMENT-RANGE). The Agent or causer shows up in sign as the active participant,

the patient as the usually stationary participant being acted upon. As in relational grammar, these relations are based upon observational properties of the terms with respect to the verb. The relational model is attractive because it does not force one to specify the syntactic form of the sentence through a rigid ordering or tree structure.

Even more flexible is a frame analysis model which allows one to speak in terms of a scene or visual image. Proximal relations can then be preserved without translation into any linear forms. The frames approach emphasizes an important aspect so often repeated in descriptions of ASL. What one is doing is building a picture -- a scene. The signer is always thinking in terms of the picture he is presenting. He is trying to produce a miniature characterization of a real event. When elements of the event are present and within access for him to refer to in his discourse, he will use them. For example, he will point to an actual person rather than producing an arbitrary grammatical index to refer to that person. Describing sign language through frames allows one to stress the visual picture being presented. It allows also for the smooth integration of other communication conventions used within the speech act. For example, if mime is found to be more explicit than the use of conventionalized ASL forms, it can easily be incorporated into the discourse making the total presentation a more direct representation of the event.

2. Visual Logic

Boyes (1972) gives various arguments based on visual perception experiments for analyzing sign in terms of visual logic. By 'visual logic,' she means a system of rules similar to the rules people use to make sense of any

visual experience. In the next section we show that frame analysis can be considered an appropriate visual logic for sign language. First we would like to present the basic arguments from Boyes (1972) for using visual logic since these arguments also support the use of frame analysis.

There are three major results of visual perception experimentation which Boyes cites in order to begin a study of the constraints that the visual mode puts on a sign language. These results all show the limitations of visual memory as compared to auditory memory. These memory processes can each be divided into the same three stages. First, there is the initial storage of the stimulus which is identical to the actual stimulus. This part of memory is referred to as iconic memory (visual mode) or echoic memory (auditory mode). The next stage is short term memory where rehearsal can take place. Rehearsal is the process of repetition of the stored material during which the material is decoded, i.e., grouped into meaningful segments. This recoded material is then stored in long term memory.

One result that Boyes cites is that iconic memory is shorter than echoic memory. Iconic storage usually lasts for between 250 msec and 1 sec whereas echoic storage can last as long as 10 sec. A second fact is that the reaction time to visual stimuli is longer than that to auditory stimuli. The third result is that visual short term memory is more limited than auditory short term memory in that it does not seem to be able to hold as many items in the presence of continued input. The current figures for this are 4 or 5 items maximum in visual STM as opposed to 7 ± 2 items in auditory STM. Boyes claims that this difference is due to the limited capacity for rehearsal of visual information.

All three of these results show that there is generally less time avail-

able for processing the sign sentence then there is for the spoken sentence. The temporal segmentation of sign would have to produce segments short enough to fit in iconic memory. And the sentence would have to be structured in such a way as to not tax STM with its limited rehearsal capacity. The sentence structure cannot rely on dependencies of elements which are temporally separated beyond the span of visual STM. Boyes seems to go a bit too far here and says that there should not be a "syntax which depends on decoding a temporal succession of images as a unit." But all this really means is that the sentences in ASL must be shorter than 5 items or that they must be processed in a way that does not require linguistic links between items which are separated by more than 4 items. Of course, more must be known about the linguistic processing of sign language before these conclusions can be made more specific.

In any case, it is clear that more information must be encoded per time interval in a visual language than in a spoken language, if we assume that the rate of transmission of information is to be the same in both. This can be accomplished by the mode of production in two ways. First, the symbol system used must be more direct, i.e., there should be a simpler mapping between visual sign and meaning than there is between sound and meaning. Secondly, sign must utilize its spatial dimensions to overcome the temporal limitations on the transmission of information. Frame analysis is able to represent these qualities of ASL.

3. Frame Analysis

Frames are a convention for representing knowledge. Frame analysis is a method for representing language as a system of frames. There are four

different types of linked frames that we will be using. These are discussed in Thompson (1975). Thompson attempts to resolve the apparent conflict in terminology with reference to the notions of scenes and frames in the work on prototype semantics (Fillmore and Rosch, MSSB, 1975) and the work on natural language understanding systems (Winograd and Bobrow, MSSB, 1975). In order to do so, he focuses in on two dichotomies. The first yields two types of frames, those representing knowledge of events and those representing linguistic knowledge. The second dichotomy further refines the categorization so that each type of frame can describe prototypic knowledge or knowledge of the instance at hand. These distinctions, then, give rise to four types of frames: Scene Prototype Frames (SPF), Scene Instance Frames (SIF), Linguistic Prototype Frames (LPF), and Linguistic Instance Frames (LIF). Before we discuss the structure of each type of frame we would like to indicate their possible functions in processing ASL. A sees an event and an SIF is formed with guidance from the appropriate SPF which was activated when one of its principle defining characteristics had been recognized. A wishes to communicate this scene to B. A constructs the sign sentences by following the links from the SPF to an LPF. The LPF will guide the filling in of an LIF based on the actual participants in the SIF thus producing the appropriate sign sentences. B watches A's signing and essentially reverses this process. An LIF begins to be formed and activates an LPF which guides the filling in of the LIF and causes the activation of an SPF. The SPF guides the filling in of the SIF with information from the LIF. Once the SIF contains all the requisite information, B is said to have understood what A signed to him.

What information do these frames contain and what are the various links, or "perspectives" as Thompson calls them, between these frames? Thompson

suggests a certain internal structure for these frames.

> A frame contains at least three sorts of things: slots, states, and actions.
>
> Slots are for identifying the participants in a given frame. Each slot has a name and a value. In an Instance Frame, these values will usually be names of other Instance Frames which describe the things which are filling each slot, while in Prototype Frames, they will usually be names of other Prototype Frames which contain information about the sort of thing which can fill the associated slot.
>
> States are statements about various relationships which hold among the slots, and actions describe transitions between states.

We will need a slightly different structure because of the kind of information that is usually presented in sign. The major addition that we make is a category of slots called Ground which contains such things as the setting and the time element. We call the rest of the slots Figures. An example of an SPF would be {PREDITOR-PREY}.

{PREDITOR-PREY}

Slots

Ground	Figures
TIME {time}	PRED {animal}
PLACE {place}	PREY {animal}

States

I. PRED doesn't have PREY
II. PREY has protection
III. PRED gets PREY
IV. PREY gets caught

Actions

A. I. becomes false and III. becomes true
B. II. becomes false and IV. becomes true
C. I. becomes true and IV. becomes false
D. II. becomes true and III. becomes false
A or C, A implies B, C implies D

An instance of this frame would have the ground and figure slots filled in with links to other instance frames as in the following SIF..

{PREDITOR-PREY}

Slots
  Ground
    TIME {narrative time 413}
    PLACE {house 584}
  Figures
    PRED{wolf 02}
    PREY {pig 98}

States and Actions (as in SPF)

The corresponding LPF would look much the same except for the crucial addition of the verb. An LPF contains Ground and Figure slots along with a verb slot. The States and Actions are no longer present. Presumably the verb and the cases encode all this information. A perspective is given in order to match the Figure slots in the SPF with the case slots in the LPF.

{PREDITOR-PREY}

Slots
  Ground
    TIME {position on time line}
    PLACE {position in sign space}
  Figures
    AGENT {'animal'}
    PATIENT {'animal'}
    VERB {'lex WANT,GET,EAT'}

Perspectives
  {PREDITOR-PREY,SPF}
    PRED = AGENT
    PREY = PATIENT

This account of the LPF is much in the spirit of Thompson's LPF. But our account of the LIF is different. We are dealing with sign and not a spoken language. The case relations are clearly manifested on the surface in sign because the hands act out the scene. So our LIF looks as follows:

{PREDITOR-PREY}

Slots
  Ground
    TIME {position on time line 617}
    PLACE{position in sign space 729}
  Figures
    AGENT{wolf 44}
    PATIENT{ pig 91}
    VERB {WANT,GET,EAT}

There is no need to have Thompson's perspective to tell us what case roles

the subject, object, etc. of the verb play in the prototype. Processing will be faster since the linguistic prototype and instance frames are more alike in ASL.

In sign the four frames are more alike in structure and there is much less need for links between frames. This cuts down processing time greatly and compensates for the limitations on visual memory. Linquistic frames differ from scene frames in the presence of the verb. As Reid says, the grammar of the image is different from the grammar of the language in that the image is made up of participants and properties attributed to them whereas the sentence is a package held together by the verb. Frame analysis formalizes this notion and reflects the speed of processing ASL. We propose that it be seriously explored as a linguistic theory for sign language.

4. A Frame Analysis of Sign Language

The remainder of this paper will include a description of some devices in sign as well as a discussion of how they might be handled by a theory of Frame Analysis. These devices are not only interesting features to analyze, but also reveal the structure of the frames (focus, boundaries, weak points).

Indexing is a process in ASL which parallels pronominalization and deixis (this, that, here, there) in spoken language. There are two types of indexing: real world references and conventional references.

Real world references are of the type discussed earlier. When the person referred to is in the vicinity, one points directly to that person rather than to an arbitrary index. The same goes for location. Also, a person recently having left a group of signers will be referred to by pointing to the position he previously occupied.

In frame analysis, the grammatical to real world reference link could be achieved by resorting to a higher frame encompassing the speech act. This speech act frame monitors the entire event and specifies what is common knowledge shared among the participants in the speech act. That shared knowledge determines the set of objects, persons and locations which can be referred to directly (by means of pointing). For example, if A knows that B has in his knowledge of the room they are in the vision of a bookshelf in one corner, then A can point directly to it without having to name it. The same goes for the shared knowledge of locations. If two people share the knowledge that city X is the obvious referrent of a point back over the left shoulder, then it will be used. Where this knowledge isn't shared, this referencing would be forbidden.

There are several types of conventional indices for things, locations and people as well as positions for such indexing. The stationary person index, commonly referred to as grammatical indexing, involves referring to certain individuals by pointing to conventional places within the signing space: right, left, distal right, distal left, and straight ahead, in that order (for a right-handed signer). Indexing into these positions allows ready reference at any following time within the discourse.

Grammatical indexing uses a frame for reference similar to the speech act frame. In this frame, however, index points are specified as to which arbitrary referents are tied to them. In cases where participants are closely linked to spatial locations, they use these locations as their index points.

Indices must be established (i.e. JOHN (indexed left position); ALICE

(indexed right position)). Since the tie between these indices and their referents is weak and arbitrary, they must frequently be reestablished. In the videotape, reindexing played a role in aiding us in our determination of frame boundaries. Reindexing interacts with the sign we have termed NEUTRAL POSITION (arms drop to sides). NEUTRAL POSITION is used to mark the end of a long discourse. Directly following NEUTRAL POSITION, at the beginning of a new frame, the signer would reindex 3 (the sign THREE) and focus upon one of the three pigs. Reindexing also marks mistakes and overcomplicated referencing.

Besides NEUTRAL POSITION, there is another PAUSE SIGN which aids in the delineation of discourse and, therefore, in the discovery of frames. The PAUSE SIGN occurs at breaks between actions within frames or at shifts between agentive characters in frames.

Other key sign structures which aid in frame determination are body position shifting and the use of index markers. As a result of the limited length of this paper we cannot fully examine these devices here. However,an extended version of this paper and copies of the transcription of the videotape of "The Three Little Pigs" are available from the authors.

Acknowledgements.

We would like to thank Tommy Radford for his help both in the signing of the story of The Three Little Pigs and in providing helpful comments for its analysis. We would also like to thank the Sign Group and the Frames Group from the MSSB summer meetings, Berkeley, 1975. A special note of thanks to George Lakoff whose insights into our common interests made this paper possible. The research for this paper was supported by the 1975 MSSB Workshop on Alternative Theories of Syntax and Semantics.

Bibliography.

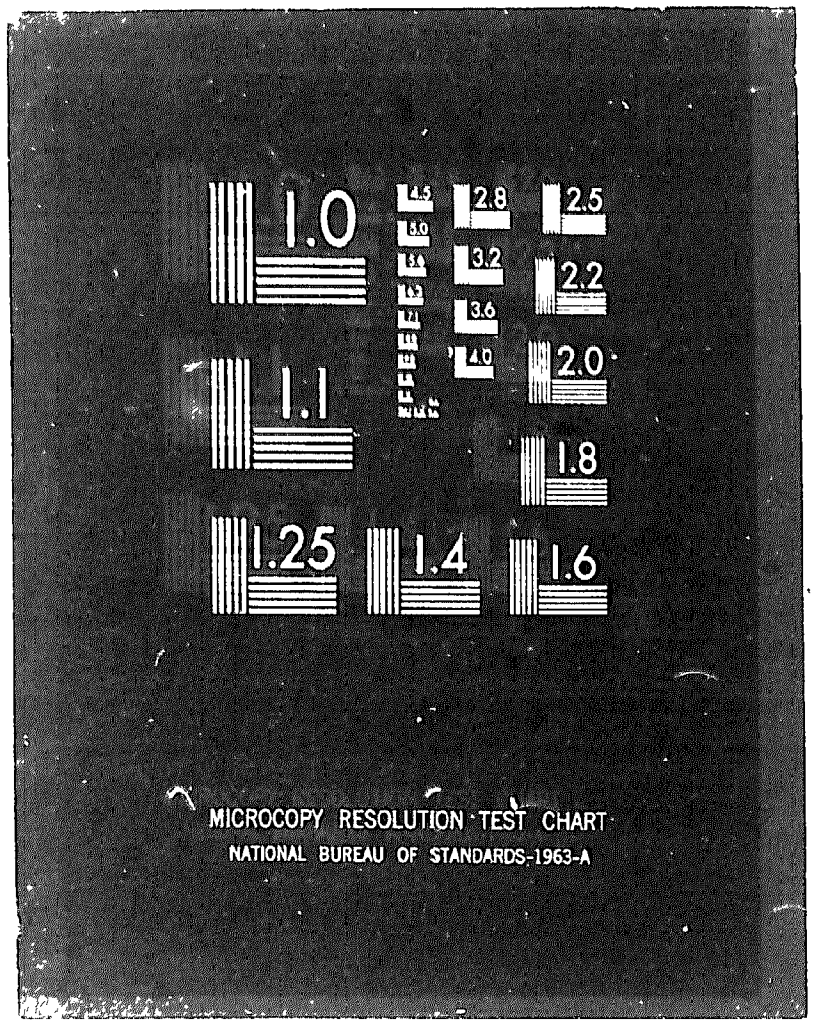
Boyes, Penny. 1972. "Visual Processing and the Structure of Sign Language." unpublished ms.

Friedman, Lynn. 1975. "On the Semantics of Space, Time, and Person Reference in the American Sign Language." unpublished Master's Thesis, University of California at Berkeley.

Reid, L. Starling. 1974. "Toward a Grammar of the Image." Psychological Bulletin, vol. 81, no. 6 (June), pp. 319-334.

Thompson, Henry. 1975. "Frames for Linguists." unpublished ms.

END


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A