MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

AI Working Paper No. 141 March 1977

# The Semantic Component of PAL:

# The Personal Assistant Language Understanding Program

by

Candace Bullwinkle

*Abstract:* This paper summarizes the design and implementation of the "semantics" module of a natural language undertanding system for the personal assistant domain. This module includes mappings to deep frames, noun phrase referencing and discourse analysis.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defence under Office of Naval Research Contract N00014-75-C-0643.

## Table of Contents

## Section 1: Introduction

This progress report summarizes the design and implementation of the "semantics" module of the natural language understanding model for the personal assistant domain called PAL. The term "semantics" here is somewhat misleading because only a portion of the work done can actually be called a semantics of language, and even that portion does not address the problem of providing the semantic knowledge needed for parsing a sentence.

In a breakdown of the overall language system design given in diagram 1, the parser uses its own semantic database (which is implemented as semantic markers) to assist it in parsing. It passes its result to a case-frame mechanism which builds Fillmore [1]-like case frames for use by the deep semantics. The parser, shallow semantics and case-frame mechanism are more fully described in Marcus [2]. The concern of this paper will be the deep semantic component, which has the purpose of providing the pa-scheduler with a series of frames that represent the speech act and knowledge of the input sentence. In the remainder of this paper I will be assuming that the reader is familiar with the frames representation developed by Goldstein and Roberts [3], and I will refer to these frames as deep frames to distinguish them from case frames.

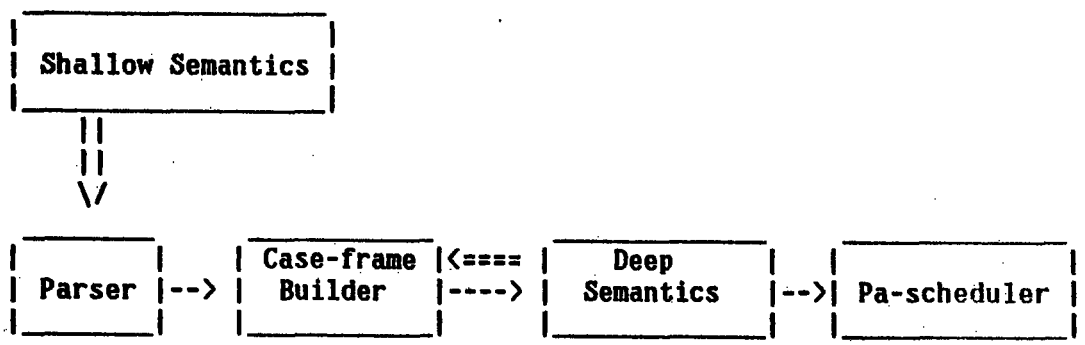


Diagram 1

One last comment must be made about the overall design of the system. This system was designed to make use of the notion of pipelining information through the system. Thus as the parser completes some portion of the sentence parse, it passes its partial results to the case-frame builder. The case-frame builder also passes to deep semantics messages about what it has constructed as it constructs each piece. Deep semantics passes its results about each sentence to the pa-scheduler without waiting for the whole discourse of the user to be completed. This pipelining effect was originally included to assist each person developing a part of the design to maintain some modularity and independence from his/her co-designers. In fact it has turned out to provide much modularity especially for error recovery and debugging. Use of pipelining implies independent control of the system modules, resulting in a control structure that may make use of parallel co-operating processes. To this time, no parallel control has been introduced.

The deep semantics consists of four separate modules. The components of deep semantics are summarized in the flowchart below.

CIFB ==> Referencer ==> Sentence-level pragmatics ==> Discourse analysis

The first is the case to independent-frame builder (CIFB). Briefly this component maps case-frames into deep frames which can be use by the database. It interacts (hence the double arrows in diagram 1) with the case-frame builder for additional information about the parse tree which the case-frame builder does not tell the semantics of its own accord. The result of the CIFB is handed to a referencing module which chooses referents for all the frames. These referents may be frames from the previous discourse or frames previously stored in the database. A special job of the referencer is proper handling of time phrases. The referencer is described in Section 3.

Referencing is followed by two modules which contain rules associated with the pragmatics of natural language {note 1}. The first is a small module of rules associated with knowledge communication on an intra-sentence level. More about the sentence-level pragmatics is given in Section 4. The other pragmatic component builds a discourse model and determines the speech act of the sentence so that the scheduler is given the proper task to do. The discourse structure is needed especially for multi-sentence interaction. This module is described in Section 5.

The above model is a simplification of what I believe to be a more complete model of natural language understanding in any domain. With more sophisticated demands on the shallow semantics (for which semantic markers could not discriminate), the semantic knowledge needed to parse a portion of a sentence may require processing by the case-frame builder and deep

semantics (even as completely as including processing by the discourse module). However, this simpler model allows for experimentation on how to build and understand deep level frames once the sentence parse is well specified. The more complex model will be needed so that a system such as this one may distinguish word senses.

Two hypotheses are suggested by this implementation effort. First, a deep frame based system for natural language allows much of the knowledge coded as independent LISP programs to be associated with individual frames and thereby localizing many of the language rules. This hypothesis is one of Minsky's [4] original claims for the use of frames. Secondly, via the procedural methods associated with deep frames, it is possible for slots filled by the case frame mechanism to automatically fill slots needed by the pa-scheduler. I say "possible" because it is now clear how to do this after the past several months of writing the CIFB. This suggests that the notion of a different viewpoint within a frame can by handled effectively by procedural methods.

## Section 2: The Case to Independent-Frame Builder

The deep frames built by the CIFB contain all the sentence information in a form usable by the pa-scheduler. Thus, for example, the frame for "a meeting with Ira" given below includes not only the determiner (under self $det) and the head noun of the noun phrase (under self $head), it also contains the node in the parse tree for the phrase (under self $node) as well as the semantic relationship of Ira to the meeting (under participant

$val) and the node for the noun phrase "Ira" (under participant $lv). With this structure, the deep frame has access to any information in the parse that it might need for later processing or error recovery. The ref and script slots will be discussed in Sections 3 and 5 respectively.

```
(FASSERT FRAME3
        (SELF ($DET (A)) ($HEAD (MEETING)) ($NODE ((NODE6)))
        (SCRIPT ($EXP ((ASSERT PLACE))
                      ((ASSERT WHEN))
                      ((ASSERT PURPOSE))
                      ((EXPAND PARTICIPANT)))
                ($EXP-MET ((ASSERT PARTICIPANT))))
        (AKO ($VAL (MEETING)))
        (PARTICIPANT ($LV ((NODE23 .  1)))
                     ($VAL (IRA (IN:  FRAME3))))
        (REF ($VAL (NEW))))
```

The CIFB also handles some cases of synonomy. Since the pa world is relatively small, there are only a few primitive actions. The CIFB is capable of seeing some verbs as synonomous to a given pa-primitive verb by means of a mapping. The CIFB also translates the case-frame slots of a verb to their equivalent pa-primitive slot form. Thus for the English verb "schedule," the case-frame slot "agent" is translated to the agent slot of the primitive action "schedule" of the pa domain.

Another aspect of the CIFB is the means of control passing between it and the case-frame builder. The CIFB accepts messages from the case-frame builder, processes the message and returns control to the case-frame builder. In general the messages sent by the case-frame builder contain only enough information for the CIFB to know which structures exist in the parse and case-frame. While processing a message, the CIFB may decide it needs more information than is specified by the message. It can request

this information of the case-frame builder by simple function calls. Some functions allow the CIFB to ask questions about pieces of the parse tree while others may request information about the case-frames themselves.

Message passing with the ability to expand the message via access functions introduces some difficulties for debugging. The caseframe builder must not pass a message to the CIFB until all the questions the CIFB may want to ask of the caseframe have either been computed or can be computed from existing structures. Should a message be passed from the case-frame before this, the CIFB may request unavailable information. For example, the CIFB may request the value of a node which the caseframe has announced as being in the parse tree. If the parser has not attached the node to the tree before the message was sent, the CIFB lacks needed information and cannot continue. The CIFB cannot accept "I don't know" messages because it does not have a way of discovering when the attachments have occurred. Producing a proof that all messages are cleared, even informally, is a formidable task and has never been resolved successfully in this implementation. The lack of such proofs often means introduction of bugs between the system modules.

A major subtask of the CIFB is the handling of time phrases. Use of time phrases requires several design citeria:

(1) a time representation must be chosen that is adequate for all the different ways time phrases can be stated in English.

(2) the representation must be compatible with the operations carried out by the scheduler.

(3) the representation must be useable when incomplete specification

of time is given.

(4) predications such as "a meeting is before 3 p.m." must be expressible in the representation.

(5) the representation must allow for embedded time phrases and as well as other kinds of embedded noun phrases, e.g. "the 3rd day of the 4th week of May," or "the week of the party for John."

At a very early point in the pa-design, Goldstein and Roberts [3] used as a time representation a single point in time. This was quickly seen to be inadequate especially for criteria (1) and (5). Instead they chose a calendar representation for time, consisting of a year, month, day, hour, and minute. The calendar representation was expanded into the notion of a frame for time when the final language design was undertaken. This switch was useful from a language design viewpoint because incomplete specifications such as "Tuesday" or "the day after the party" were expressible in a more direct fashion in the frame representation than in the calendar representation.

Once a frame was chosen as the representation for time, a final design decision was needed. The choice of a necessary and sufficient set of slots for the variety of English forms was needed. Thus in addition to year, month, day, hour and minute slots, Levin [5] and I determined that weekday, time of day and week slots were desireable. While the information specified by one of these could have been handled via the requirements facet of another slot, specifying slots for weekday, time of day and week slots makes the information more perspicuous. In the frame example below all the slots are used for the phrase, "next week on Friday, October 29,

1976 at 5 p.m." In the example, the reader will note the use of a special procedural form in the slot of "week". More about these functions can be found in Goldstein and Roberts.

```
(FASSERT FTIME21
         (YEAR ($VAL (1976)))
         (MONTH ($VAL (OCTOBER)))
         (DAY ($VAL (29)))
         (WEEKDAY ($VAL (FRIDAY)))
         (HOUR ($VAL (5)))
         (MINUTE ($VAL (0)))
         (WEEK ($VAL (@ (calendar-week (hence 1 'week (now)))))
         (DAYTIME ($VAL (P.M.))))
```

Time phrases are processed in a two phase control where the CIFB builds a partial representation of time phrase (see diagram 3) and then a special time referencing capability completes the translation into a form needed by the pa-scheduler. Time referencing is necessary because the CIFB representation is ambiguous. The time referencer operates as part of the reference capability. In the diagram, the phrase "Thursday, October 28" is described by the frame ftime50. The finterval46 and relation47 frames describe the relation between the time phrase and the event it is associated with. Finterval frames have slots for start and stop times of events. The time slot of finterval46 will be associated with one or perhaps both slots once the entire sentence is processed. It is the job of the time referencer to make the correct association from partially instantiated frames {note 5}.

```
(FASSERT FINTERVAL46
         (AKO ($VAL (FINTERVAL)))
         (TIME ($IV (RELATION47))))

(FASSERT RELATION47
         (AKO ($VAL (RELATION)))
         (SELF ($HEAD (DURING)))
         (ARG ($LV (CASEFR40)) ($IV (FTIME50))))

(FASSERT FTIME50
         (AKO ($VAL (FTIME)))
         (MONTH ($IV (OCTOBER)))
         (DAY ($IV (28)))
         (WEEKDAY ($IV (THURSDAY))))
```

Frame resulting from CIFB processing on the phrase "Thursday, October 28"
Diagram 3

The two pass control of CIFB and time referencer was chosen because of the following characteristics:

(1) A single time phrase can be used for multiple specifications of time. These uses are difficult to collect in a single pass. For example, the phrase "Thursday" in "the meeting on Thursday from 3 to 4" is a specified time for both the beginning and end time of the meeting.

(2) Phrases embedded in a time phrase, e.g. "the Third day of the $th week during John's visit," particularly with a predication, require one pass to describe the strucutre and one to flatten out the embedded structure into an ftime frame.

While other objects in the pa domain, such as locations, may also have a complex representation, consideration of time phrases and the designs made regarding its representation have given the designers a guide to the representation of these objects.

One considerable limitation on the time representation is the lack of ability to use quantification. Such phrases as "every Tuesday in May, sometime next week" or "any Thursday in 1976" cannot be represented currently. For the universal distributive quantifiers "every" and "each," there are clear representation possibilities, but for "some" and "any", the choice of representation requires further study.

From this discussion it may be clearer why the CIFB is called the Case to Independent-Frame Builder. It is the job of the CIFB to translate case information to frames, but these frames are still independent of the pa-scheduler and its actions.

## Section 3: The Reference Module

The reference module distinguishes four kinds of reference found in English usage:

(1) pronominal reference: for pronouns like "he" and so on.

(2) proper names

(3) discourse dependent noun phrases: a noun phrase with a referent somewhere in the previous discourse, e. g., "the book" when some book has been previously mentioned. Discourse dependent noun phrases include anaphoric referents like "the one" and the referent of "my" in "my sister." While pronominal reference is similar to other discourse dependent phrases, it is distinguished from these because of the need for some special intra-sentence rules governing its bindings (note 2).

(4) discourse independent noun phrases: a noun phrase which has not been previously mentioned in the discourse but is assumed known to the system. For example, "Candy's office" where the referent of "Candy" is known, has a referent that need not be stated in the discourse for this phrase to be well specified.

The reference module is designed to determine referents for these four classes of nouns. It success varies largely due to the need for more rules about disjoint reference than have been included in this implementation. The reference module relies upon a function called FQUERY [6] to find frames that match a given description and also upon a discourse model that narrows the possible topic of conversation to a small collection of frame objects.

The description for FQUERY is a special frame called a relation frame. This frame explicitly states the semantic and knowledge constraints on the referent of the noun phrase. Thus a simple noun phrase like "the book" will have a relation associated to it that says that the referent is a kind of book while "my meeting with Mitch" will be the conjunction of relations that the referent be a kind of meeting, and that the participants in that meeting include Candy and Mitch.

In addition to a description, FQUERY can take a universe over which it should search. For noun phrases in the discourse dependent class, the noun phrase must have a referent in the discourse or be associated to one of the expectations of the discourse (see Section 5 for discussion of expectations). For noun phrases in the discourse independent class, the referent may be in the discourse, or it may be in a larger context which

can be grown from the discourse. The need for a discourse universe is evident even in the small pa domain, which already has sufficient people, places and events to make searching the entire database unacceptable.

There are presently some limitations on the use of FQUERY. FQUERY permits no use of existentials and hence this must be simulated by first finding a class of objects of the existential type and then calling FQUERY with each object individually as part of the relation description. Furthermore, FQUERY could be "made smarter" by eliminating the need for the explicit description given in the relation frame. It is possible to design FQUERY to assume directly from a frame for "Candy's office" that the referent ought to be a kind of office and have Candy as possessor of that office since the frame for "Candy's office" contains these descriptions in a clearly identifiable way.

When finding the referents of pronouns, the referencer again makes use of the universe capability of FQUERY. However, before choosing the discourse context as the universe a relation exists over, just the frames in the current sentence may be used, so that intra-sentence pronominal references can be found. In some ways the database does not have an adequate description for pronominal referencing. Currently the ako heirarchy makes no male-female distinction, which means that gender markings on pronminals cannot be completely used. The pronominal referencing is also limited because many of the intra-sentence rules regarding pronouns have not yet been introduced.

Finally, the referent module has a special capability for determining the unique referent of a proper name. The parser gives a special

representations of names of individuals which distinguishes titles like "professor", "mrs.", last names and other names. This information is given directly to a program which searchs through the people in the database and returns a list of unique database atoms for all possible referents. When this list is longer than one, the system must ask the user for a referent. However, given the discourse ability to define of topic of the discourse, ambiguous referents may be distinguishable on a pragmatic basis. For example, the referent of "Chuck," given the theme of "pa meeting" could be found by considering the default participants of the pa meeting while when the topic is "dinner party for the Watergate gang," the default participant is likely to be a different Chuck.

While the reference module has some reference capabilities, it should be clear that there is much work to be done before very much complexity in referencing can be handled adequately. It is my belief that by expansion of the various sentence level rules and by the use of the discourse structure, the reference module may be able to handle a class of referents not understandable in A.I. programs before.

## Section 4: Sentence level Pragmatics

Many sentences in the pa domain need to make use of information that is neither syntactic or semantic. Most such information is related to the discourse context in which a sentence is communicated and hence is inter-sentential pragmatics. However, some rules apply to individual sentences and it is these rules which are included in the sentence level pragmatics.

Thus if a user says S1, then it can be reliably assumed that the referent of "I" is a participant in the meeting.

(S1) I want to meet with Ira.

Even if the statement doesn't include mention of the speaker, the speaker may still be assumed to be a participant. In interpreting S2, the speaker is likely to be considered a participant.

(S2) Schedule a meeting with Ira for Monday.

The rules for associating such information with sentence input must make use of both case-frame and deep frame information. The rules are associated with slots of the deep frame but may require certain tests about case-frame structure. For S1, a pragmatic rule about modalities with complement structures "meet" and "schedule" causes the actor of wanting to be assumed as a participant in the meeting. For S2, a rule associated with "schedule" as an imperative concludes the speaker is a participant in the activity provided there is no benefactive case (this rules out S3 below).

(S3) Schedule a meeting for Ira with Jim for Thursday.

So far, all sentence level rules are simple local constraints on the verb of a particular sentence. No theory of these rules has been assumed or is being proposed. With further development, it is hoped that enough rules will appear to suggest some model of communications between speaker and hearer.

## Section 5: Analysis of Discourse

In understanding the multi-sentence discourse of a person

communicating with natural language, a context that describes the discourse is useful. With this description, each new sentence of the discourse can be related to the past sentences of the discourse. New sentences must be related to the previous discourse for two reasons. First, the references of the phrases used in the current sentence may be located in a previous sentence. The context can provide an information structure which can be indexed quickly for previous referents. The discourse context also distinguishes those items that are part of the discourse from the rest of the objects in the database.

Another reason for discourse context is for specifying the speech act context of the current sentence. It is easy to see that the request (or lack of one) conveyed by the use of a particular sentence varies as the context changes, as the second sentences of D1 and D2 show. To distinguish the speech act context, it is necessary that the previous context be available so that its characteristics can be computed as needed.

D1 (secretary to pa-system): Kurt wants to schedule a meeting for Tuesday. Since he has classes at 2, the meeting needs to be scheduled for 3 or later.

D2 (story): Bill and Dave want to decide when to have an electronics club meeting. Since Bill has classes at 2, the meeting needs to be scheduled at 3 or later. Dave, on the other hand, ...

In the pa language system, the discourse context is created from the frames built by the CIFB and which represent individual sentence input. The context consists of a frame representation of objects mentioned in the user's discourse and includes frames for actions. It does not include

frames that are references to other frames in the discourse, since these reference frames are redundant information. Thus a frame representing the phrase "the meeting" is excluded from the discourse context when that phrase refers to another frame representing "a Thursday meeting." However, some frames that have reference to previous referent classes are included because the frame introduces a new object. For example, when a frame is a subset of another frame, the frame denoting the subset is included. Thus in D3 below "the meeting on the 14th" is included in the discourse context because it bears a subset relation to its reference class "two meetings in May."

D3: There are two meetings in May. The meeting on the 14th includes John and Phil while the one on the 3rd has Pete and Ira attending.

To integrate a new sentence to the discourse, two other pieces of information are needed: the topic pointer and a possible speech act list. The topic pointer points to the frame that represents what the discourse is about. For example, in a discussion about scheduling meetings, the topic pointer points at the frame for scheduling. The possible speech act list enumerates the speech acts that a particular verb may mean in a given discourse. The possible speech act list comes from the frame of a particular verb (possibly by inheritance from higher in the tree). It includes a set of constraints that define context topics in which this verb could have the speech act type listed. Thus the verb "schedule" in a context of scheduling suggests asserting additional information about the scheduling as example D4 shows. This example seems trivially true.

However, for the modal form of "meet" to be seen as asserting information about scheduling a scheduling context must exist as D4' shows.

D4: I want to schedule a meeting with Chuck. The meeting should be scheduled at 3 p.m. We can meet in my office.

D4': David is meeting Harriet for lunch today. They can meet then because Harriet works near David's office.

Use of the possible speech act list is not sufficient because sometimes the context changes, and the same verb can suggest a new topic for the context. Consider adding the sentence "I also want to meet with Bruce on Thursday" to D4. Now the topic has changed to a new scheduling event. This observation means that a verb must not only have a possible speech act meaning that fits the context, but also the sentence containing the verb must be about the same thing as what the topic relates to, that is, the theme of the discourse. I make a distinction between the topic which depicts what the overall context is about, and the theme which depicts what one or more sentences of the context may be about {note 3}. In my programs, I have not explicitly called anything the theme, but implicitly the topic of a scheduling discourse has a theme, which may be a meeting. In D4, after completing the first sentence, the theme is "a meeting with Chuck". It is my belief that themes can change by a process I call embedding, that is, the theme can change to one of the sub-concepts associated to the theme. The theme in D4 never changes, but in D5, the theme of "meeting" is put on the shelf in favor of "the time of the meeting" and can be picked up after "the time" is fully stated. I am somewhat convinced that themes cannot be changed in other ways without

suggesting a new context.

D5: I want to have a meeting with Sam. We need a time to meet, but Thursday is out for me. Sam is also busy on Friday. If you can find a time that is okay for us, schedule it and tell Sam.

Since themes cannot be changed without suggesting a new context and since the possible speech act list only suggests a speech act given a topic, a sentence must be confirmed as being about the same thing as the theme. Thus in D4, sentences two and three must be about "a meeting with Chuck" as well as having the speech act content of an assertion. Sentence 2 is trivially so since it explicitly refers to the theme. However for sentence three of D4 to be about "a meeting with Chuck," there must be an assumption that meetings and places are related.

To capture these assumptions, I have used the notion of an expectation. In the pa world mention of meetings raises expectations about assertions of place, participants, times and purposes. In scheduling meetings, assertions suggest definite assertions about aspects of a meeting whereas in defining meetings the assertions say more about default preferences. Therefore, associated with a meeting are expectations that say what things are likely to be asserted about a meeting while the suggestive differences of defining versus scheduling rests in the constraint on the verb to its possible context topic.

Currently, expectations are stated explicitly in a frame as a result of filling slots. This method, however, is too example specific and was chosen only to experiment with referencing when expectations are avilable. More generally expectations should be deduced from the slots of a frame and

the primitive action being performed. Thus when scheduling a meeting, the pa-scheduler knows that it is required to fill certain slots of the meeetng frame. Using this knowledge to deduce that the user can expected to say something about these slots, the discourse module does not need the information stated redundantly on the frame.

The use of expectations is not exactly equivalent to Charniak's demons because the expectation does not have the right to fire on incoming information while a demon does. The tighter control of expectations is needed because expectations are considered only after a speech act meaning is associated to the verb. Without the speech act meaning there is little purpose in choosing expectations.

The notion of a theme as described here may also be useful in referencing although this capability has not been implemented. Mention of "the place" when the theme is some particular meeting, say meeting25, suggests that "the place" is the place of meeting25 even though the whole discourse may include several mentions of meetings different from meeeting25. This notion needs to be explored further before it can be determined whether it helps or confuses the referencing process.

There are some weaknesses in the current implementation of discourse contexts, topic and theme. Firstly theme is implicitly stated, because the real power of this notion is as yet unclear and rules for how it is chosen do not exist. By making it explicit, I can begin to explore just what is to be gained and lost by its use. Secondly there is no way to change the current topic when the discourse changes. This is in part due to not fully understanding how to recognize such changes and also due to limitations on

the ability of the pa-scheduler to respond to such a capability. Recognizing changes in the topic is related to recognizing changes in the theme. As I indicated, for D5, the theme can be changed via embedding to a concept that is related to the themes. With this new theme the discourse can proceed for one or more sentences before the old theme is re-established as the theme. However, in the case that the new theme is discussed for several sentences, the topic often changes too. This in D5 the topic is scheduling with theme of meeting. When the theme becomes time, the topic is more of a definition of unacceptable times. How all these changes can be recognized is still unclear.

I also have not discussed how to recognize the initial topic. In fact as I discovered some time ago (note 4), this recognition problem is extremely difficult in the general case. In the pa world, it appears that the first sentence of every discourse is an immediate cue to the topic in much the same way as topic sentences of paragraphs are in a well written paper. Without this cue, interpreting the user's request will be extremely difficult.

Finally much more testing of the discourse module is needed before it can be ascertained what limitations, shortcomings and capabilities this module provides both to the pa-scheduler and to the referencer.

## Section 6: Where to from here

In the last section I indicated a number of directions for experimentation with the discourse module. I also commented that the notion of theme may be useful in reference determination. Should that be

the case, further study of the relation between theme and the discourse context which the referencer currently uses will be needed as well. In addition to the comments on discourse in Section 5, one more area that needs to be explored is the effect of tense on establishing the relation between a sentence and the discourse context. For many examples past tense appears to be a dead give away that a scheduling request is not being made.

In the CIFB, much work remains to be done to properly include modifier type clauses including restricted relatives in the CIFB's capability. Choosing a reasonable frame representation for the restricted relatives is a significant sub-problem of this work. Finally inclusion of ambiguity and word sense to the CIFB remain to be done.

In the reference module, rules need to be explored for intra-sentential referencing as well as expanding the anaphora capabilities of the module in many ways. The referencing module needs to be expanded to include the situation where the reference is ambiguous until after the discourse context is considered fully.

Finally, the intra-sentential pragmatic rules can be expanded for many examples. As I indicated no coherent theory of this component has yet evolved, but with attention this component may be expandable to something less ad hoc or perhaps eliminated in favor of another method.

Notes

{note 1} By "pragmatics" I mean those parts of language understanding that involve rules of communication and knowledge of one's environment which are independent of the particular natural language the speaker understands and uses.

{note 2} Examples are rules such as Ross' [7] Backward Anaphora Constraint

and Lasnik's [8] reformation of BAC.

{note 3} I mean here by theme a notion similar to that of Kuno [9]. While Kuno's definition is even vaguer than mine, his examples are insightful and convinced me that the notion ought to be explicit in my work.

{note 4} The thrust of Bullwinkle [10] was to recognize the topic of simple stories for children.

{note 5} The use of a $IV facet is an extension of the other frame facets. It is used instead of the $val facet because filling the $val facet causes if-added methods to be run. Since the time referencing is handled in a two pass control, filling a $val during the first phase would run the if-added methods too soon. Hence $iv is used for values inserted by the first phase of time referencing.

References

[1] Fillmore. "The Case for Case," in *Universals in Linguistic Theory*, Bach and Harms (eds.), New York, 1968.

[2] Marcus, Mitchell. Unititled doctoral dissertation, M.I.T. <forthcoming>.

[3] Goldstein, I. P. and R. B. Roberts. *NUDGE, A Knowledge-based Scheduling Program*, submitted to IJCAI-77.

[4] Minsky, Marvin. "A Framework for Representing Knowledge," *The Psychology of Computer Vision*, Winston (ed.), McGraw-Hill, Inc., 1975.

[5] Levin, Beth. Personal discussions on time references, 1976.

[6] Clinger, William. Personal discussions on FQUERY, 1976.

[7] Ross, J.R. *Constraints on Variables in Syntax*, unpublished MIT doctoral dissertation, Cambridge, Massachusetts (1967).

[8] Lasnik, Howard. *Remarks on Co-reference*, Linguistic Inquiry, volume 2, number 1, 1976.

[9] Kuno, Susumo. "Three Perspectives in the Functional Approach to Syntax," *Papers from the Parasession on Functionalism*, Grossman et al. (eds.), Chicago Linguistics Society, 1975.

[10] Bullwinkle, C. L. "Computer Performance of the Sentence Completion Task," M.S. Thesis, University of Pittsburgh, 1975.