Parsing into Discourse Object Descriptions[*]

Lars Ahrenberg
Department of Computer and Information Science
Linköping University
S - 581 83 Linköping

ABSTRACT

This paper reports work on the design of a natural language interface with a limited dialogue capability. It is argued that (i) The interpretation of the input is preferably represented as a structure of Discourse Object Descriptions (DODs); (ii) The DODs must be determined on the basis of different types of knowledge such as grammatical knowledge, object type definitions and knowledge about existing discourse objects and their discourse status; (iii) The different types of knowledge are stored separately but integrated in the interpretation process which is based on constraints.

INTRODUCTION

The LINLIN-project is concerned with the development of general-purpose natural language interfaces (NLIs) to computer software with special emphasis on communication in Swedish. A useful general-purpose NLI must meet a variety of requirements, a number of which concern communicative ability. The communicative abilities of an NLI are necessarily restricted by the limitations of existing techniques, but can also be purposely restricted to enhance transparency. It is not certain that the linguistically more competent NLI is the most useful one, e.g. if its behaviour appears idiosyncratic to the user. In any case, the language of an NLI is a language designed (and is in that respect not a natural language) so there are many questions to be answered about how it should be designed, both in terms of how it should function as a vehicle of communication and in terms of internal representations and procedures.

As for the first aspect we are conducting a series of simulations to find out what communicative abilities an NLI should have (Dahlbäck&Jönsson, 1986), but meanwhile we are assuming that LINLIN should meet the following demands: it should have a fair knowledge of the structure of Swedish words, clauses and utterances, an extendable lexicon, an extendable knowledge of object types, an ability to cope with directives, questions and assertions as they relate to the current background system(s) and a restricted ability to engage in a dialogue with the user.

The dialogue capabilities of LINLIN are primarily designed for the following purposes: (a) to enable the user to make explicit and implicit cross-references between utterances, e.g. by using pronouns and ellipsis; (b) to allow the user to build commands incrementally; (c) to ask the user for clarifications and other information that the system might need, and (d) to provide help for the user.

In this paper some consequences of these demands for the representation and interaction of various types of knowledge that the system needs are considered. The main ideas are the following: (1) The content of user inputs is preferably represented as a structure of Discourse Object Descriptions (DODs) which relate in various ways to objects of the universe of discourse. (2) Different types of knowledge, including object type knowledge and knowledge of the current state of the discourse must be used and integrated in the construction of an interpretation. (3) To ensure generality and in contrast to the entity-oriented parser of Hayes (1984), the grammatical knowledge is not exclusively tied to object type definitions but stored separately. (4) Knowledge about the discourse status of objects is also a kind of general knowledge that must be kept separate from object type definitions. (5) In a constraint-based parsing process the grammatical descriptions and the content descriptions can be built in tandem, sometimes with the syntax in control and sometimes with the object knowledge in control. This allows us to diminish the role of the syntactic part of the parsing to recognition of significant structural patterns, using semantic and pragmatic knowledge for the resolution of structural ambiguities such as PP-attachment.

The first background system that LINLIN will work on is a group calendar. As the pilot version of LINLIN is only in its initial stages my arguments will mainly be theoretical, while the practicality of the proposed ideas remains to be proven.

THE FRAMEWORK

Discourse Objects

Virtually anything that can be perceived as and talked about as an individual may serve as a discourse object. Thus, objects and facts represented in a database as well as the user's inputs, the commands to be executed and the responses of the system are all (potential) discourse objects. Notions such as discourse elements (Sidner, 1984) and discourse entities (Webber, 1984) have been employed to denote the entities that are "specified" or evoked by the constituents of a discourse, they and their relations then constituting the discourse model of a speaker. Hayes (1984) refers to the objects, events, commands, states (and so on) that an interface system needs to recognize collectively as "entitities". In the same vein I take the notion of a discourse object to apply in the most general sense; the universe of discourse is in principle just a collection of discourse objects. A relation between discourse objects is also a discourse object although it may also, or alternatively, be attributed to one or more of its constituents as part of their descriptions.

All discourse objects are instances of one or more object types. Thus, we allow a discourse object to be viewed from complementary perspectives. For instance, from a grammatical perspective an input may be typed as a declarative sentence, whereas from an interactional perspective it may be typed as an answer and both of these categorizations may contribute information about its content.

Discourse Object Descriptions

The information that the system has of a particular discourse object is encoded in a **discourse object description**, or DOD, for short. As discourse objects generally will have some information attached to them, we may represent a discourse object as a pair of a unique label and a DOD.

DODs have the format of structures of attribute-value pairs where the attributes represent informational dimensions, i.e. ways of predicating something of the object, and the values encode whatever information is available for that dimension. An attribute of special importance is **Instance-Of** which relates a discourse object to a type. Other attributes are generally inherited from an object type definition which occurs as part of the description of an object type. An object type definition can be viewed as a skeleton for a typical instance of that type registering the defining attributes as well as restrictions on their values. For events, such as meetings or bookings, the object type definition is basically similar to a case frame

(see figure 1). The object type definitions thus encode the system's semantic knowledge, whereas the universe of discourse encodes its world knowledge.

Label: 'Meeting

Typical-instance:

    Meeting-type: [Isa 'Meeting]
    Participants: [Instance-of: 'Set]
                   [Typical-member: 'Person]
    Time: [Instance-of: 'Time-interval]
    Start-time: [Instance-of: 'Time-of-day]
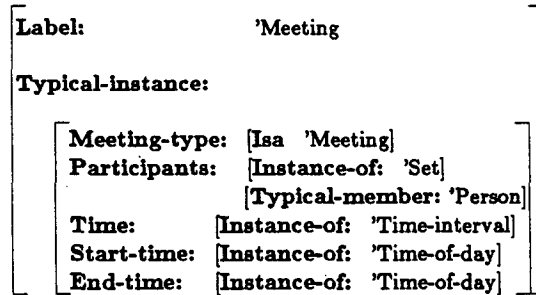   End-time: [Instance-of: 'Time-of-day]

Figure 1: Part of an object type definition.

Discourse status

We do not talk about all discourse objects at once. At any particular moment of an interaction some discourse objects are more salient than others because they are being talked about. As is well known, the way an object has been talked about at a certain point has consequences for how it can be talked about in the sequel (cf. e.g. Sidner, Webber op. cit.). It also has consequences for how other objects which are related to those salient ones can be talked about. On the other hand there are discourse objects that have a particular status in virtue of being parts of the context of utterance. Such objects are the speaker, the addressee, the time of utterance and the place of utterance. A third kind of property that distinguishes discourse objects from one another concerns whether an object is part of the shared knowledge of the actors of the interaction or not.

I will treat all distinctions of this kind as distinctions of discourse status. Objects of the first type will be referred to as *topical* and those of the second type as *central*. There can be overlap between these categories, but generally they are different. Expressions such as *my, yesterday, here* pick out central discourse objects or objects with specific relations to central objects, whereas expressions such as *his, the day before, in front* pick out topical objects or objects with specific relations to topical objects. Objects of the universe of discourse which are neither topical nor central will be referred to as *known*.

To keep track of changes in discourse status a conversational score, or score-board, is used (Lewis, 1979). One purpose of the score-board is to register topical and central discourse objects at any particular point of the interaction. This information must be updated for every new utterance. How this should be done is a difficult problem that I will not address here. However, in this area

we prefer simple algorithms to high coverage as we are not aiming for a complete solution to the problem of anaphoric reference, but for something which can be useful in man-machine dialogue.

The score-board has another important duty as well, viz. to register expectations on user input. For illustrations, see below.

Parsing and Interpretation

The entity-oriented parsing of Hayes (1984) is proposed as a suitable technique for interfaces with restricted domains. The characteristic feature of this technique is the close coupling between semantic and syntactic knowledge. Each entity definition is coupled with a "SurfaceRepresentation" of that entity, i.e. information about how such entities are expressed in linguistic utterances. Thus, each object type defines its own sub-language as it were. This has several advantages, e.g., it allows for independent recognition of entities, it makes possible the interpretation of ill-formed input and it can also be supported theoretically: the language we use for talking about people is not the same as the language we use for talking about times or locations (or for performing various types of speech acts) and this difference is not merely a difference in vocabulary but also a difference in syntax. However, Hayes makes full use of the entity-language correspondences only in top-down recognition, i.e. in the direction from object types to instances. There is no attempt at expressing syntactic knowledge at an appropriate level of generality; every single entity type has its own SurfaceRepresentation so syntactic generalizations that hold across entities are neither used nor expressed.

Tomita&Carbonell (1986), using entity-oriented parsing in the context of multi-lingual machine-translation for multiple restricted domains, propose to capture syntactic generalities by means of separate LFG-style grammars for the different languages. The grammars are kept separate from the entity definitions (and the dictionaries) at development time, but are integrated in one large grammar at run-time. This grammar, the rules of which are phrase structure rules augmented with LISP-programs for tests and actions, can then be parsed by a suitable algorithm for augmented context-free languages.

This method presupposes that the knowledge bases that are integrated don't change in the course of processing. An NLI with dialogue capabilities must not only handle syntactic and semantic knowledge, however, but also knowledge of the universe of discourse which changes with every new utterance, so a different method must be used. Such a parser/interpreter should be able to access the different knowledge bases at run-time as illustrated in figure 2.
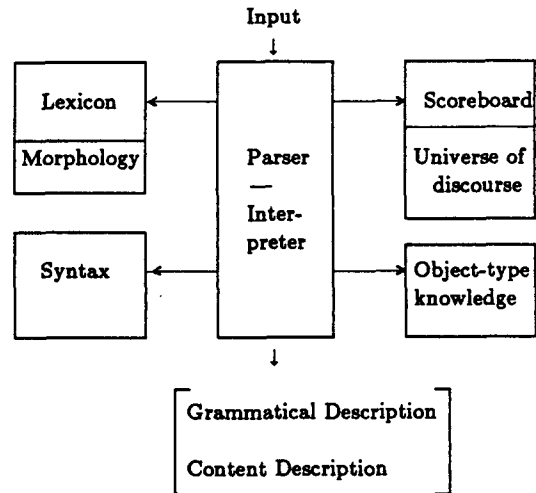

Figure 2: Knowledge bases for the parser.

The output of the parser is a DOD for the input utterance, which contains information both about its syntactic structure and its content. The grammatical description (GD) is separated from the content description (CD) in accordance with the view that they result as evaluations of the utterance from two different, but complementary, perspectives.

The content description is basically a structure of DODs. Thus, the same representation language can be used for discourse objects, object type definitions and content descriptions. Lexical entries as well as rules of the grammar are associated with descriptors which I express here as schemata in an LFG-style formalism. The construction of the content description for an input will be an incremental process, as far as possible based on unification. However, particularly in the non-syntactic part of the construction other, more complex operations will have to be used.
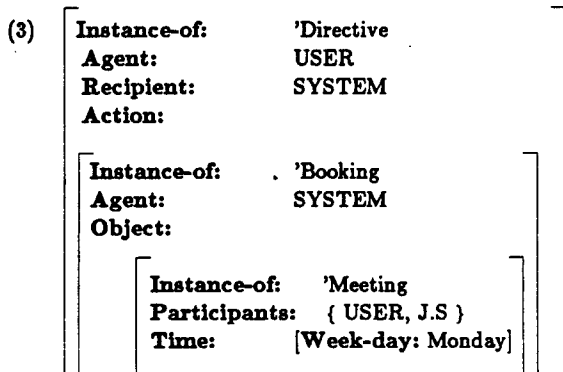
The content description can best be viewed as a contextualized semantic representation. It is partially determined by the information supplied in the utterance, but is enriched in the interpretation process by the use of the other knowledge sources. The information in the constituent DODs include (i) object type and other properties of the corresponding discourse object; (ii) the discourse status of the object, and (iii) information about identity.

142

Knowledge of the universe of discourse

Expectations - Initial hypotheses about the content description of an input may come from two sources. It may come from expectations about what is to follow or, in the absence of specific expectations, from the grammatical (and lexical) information found in the input. Utterance types are not identified with command types as there is no one-to-one correspondence between inputs and commands to the background system. Instead, inputs are regarded as messages which are classified in terms of general illocutionary categories such as assertions, questions and directives. However, many utterances will give whole or partial specifications of a command to be executed, which means that they are analysed as having that command as their topic, i.e. as (one of) the discourse object(s) that the interaction currently is about, possibly having some specific part or aspect of it as an immediate topic.

As an example, consider the short exchange below. The content description of (1) is, in abbreviated form, (3).[1]

(1) U: Book a meeting with Jim Smith on Monday.
(2) S: At what time?

(3)
```
    ┌                                              ┐
    │ Instance-of:    'Directive                   │
    │ Agent:          USER                         │
    │ Recipient:      SYSTEM                        │
    │ Action:                                       │
    │     ┌                                    ┐    │
    │     │ Instance-of:   . 'Booking          │    │
    │     │ Agent:           SYSTEM            │    │
    │     │ Object:                            │    │
    │     │     ┌                          ┐   │    │
    │     │     │ Instance-of:  'Meeting    │   │    │
    │     │     │ Participants: { USER, J.S }│  │    │
    │     │     │ Time:    [Week-day: Monday]│  │    │
    │     └     └                          ┘   ┘    │
    └                                              ┘
```

As a result of this interpretation the system introduces two new discourse objects (apart from the utterance itself): (i) a booking to be executed on the background system, and (ii) a meeting to be booked. They are labelled, say B1 and M1, and supplied with their descriptions. Moreover, both B1 and M1 are assigned topical status. The system is able to recognize information that it lacks for booking a meeting by comparing the information it has with a definition for a booking command. Having done this it may take the initiative and ask the user to supply that information, by outputting (2) above. In this case the next input from the user will be met with definite expectations,

viz. that it will be an answer relating to a topic such as <M1 Start-time>. Such expectations are registered on the score-board. They have effects not only on the content description of the next utterance, but also for the way it is parsed, as we may invoke an appropriate rule top-down, in this case a rule for the structure of a time-of-day, to see whether the expectations are met.

Another case where expectations are necessary for solving an interpretation problem is with identifications of the type (4). The form of this utterance reveals it as some sort of assertion, but there is no way of telling from the words alone what the topic is. If it occurs at the beginning of an interaction, however, it should most likely be taken as information about who the user is. In this case the expectations don't arise from a previous utterance, but from general knowledge about how interactions begin.

Knowledge about interactions is stored in the object type definition for interactions. This definition basically provides a grammar of constraints on possible interactions. The field in the score-board that registers expectations on input is maintained by a processor that has access to the interaction grammar.

(4) It is Lars.
(5) It is dry.

Topical objects - The constituent DODs of a content description must include information about which discourse object the DOD describes. Information about identity is often needed for disambiguation, e.g. to make the appropriate reading of a polysemous word. This may require consulting both the score-board and object type definitions. Thus, to interpret (5) in a system which allows *dry* to apply to different kinds of objects, say wines and climate, requires that we first identify the discourse object accessed by the subject (via the score-board topics field) and then use the definition associated with its object type to see in what way it can be specified as dry.

As a second example consider the case of PP-attachment. Wilks et al. (1985) argue (convincingly to my mind) that syntax generally fails to discriminate between alternative attachments. Instead they claim that correct interpretations can be made by a preferential approach on the basis of semantic information associated with the relevant verbs, nouns and prepositions.

However, preferences based on general semantic evaluations are not sufficient either. Our knowledge of the actual discourse plays an important role. Consider (6), which taken in isolation is ambiguous since both meetings and cancellations are objects that "happen" at definite times and therefore may be specified for time. A preferential approach must apply some ordering

---

[1] Values in capital letters are object labels obtained by special object modules. The other descriptors stem from the lexicon and the grammar (see below).

mechanism to handle a case like this. In the strategy employed by Wilks et al. the first attachment tried is to the nearest element to the left which has a preference for the content of the PP. In this case it will succeed (assuming that meetings have a preference for temporal PPs). There is an interpretation of (6) which is similar to (7), however. This interpretation is the appropriate one if we consider (6) in a discourse where the question (8) has been asked. It will also be favoured in a discourse for which there is a discourse object identifiable as 'the meeting' but no discourse object identifiable as 'the meeting on Monday'. This would be the case if there is only one topical meeting, whereas the latter expression is appropriate in a context where there is a set of meetings of the same discourse status of which only one is on Monday.

(6) You cancelled the meeting on Monday.
(7) You cancelled it on Monday.
(8) When did I cancel the meeting?

Also, the preference approach is insensitive to other global properties of the utterance. For instance, while it may be allowed to ask for information about the time of execution of a command, as in (8), and hence possible for the system to inform about it, with either of (6) or (7), it may be disallowed to request other executions than immediate ones, so that (9) and (10) would be non-ambiguous as regards attachment of the final PP.

(9) I want to cancel the meeting on Monday.
(10) Cancel the meeting on Monday.

The system can handle such cases by treating either all directives, or some subset of directives which includes bookings and cancellations, as objects that obligatorily have their temporal information determined by the time of execution. Only after they have been executed should their execution times be available as discourse topics.

We may also compare (10) to (11) and (12). Whereas (10) is ambiguous (in isolation) (11) non-ambiguously means that the meeting is on Monday, whereas (12) non-ambiguously means that the cancellation should be performed on Monday.[2]

(11) Cancel the one on Monday.
(12) Cancel it on Monday.

The pronouns must also be contextually appropriate, of course. The difference between them coincides well with the difference between the two possible interpretations of (10); (12) can be used if there is only one topical meeting

and (11) can be used if there is a set of topical meetings (cf. Webber (1984)). However, the differences in PP-attachment between (11) and (12) can be stated already in the syntax as *one* is categorized as an N that allows for PP-complements, whereas *it* is categorized as an N (or NP) that does not permit PP-complements.

Syntax and the Lexicon

It may be suggested that for an NLI the grammatical structure of an utterance has no intrinsic interest. However, most linguistic interactions involving humans seem to develop formal constraints over and above those needed to differentiate between message types and there is no reason why this should not hold for NLIs as well. Although (13) is interpretable it is not formed according to standard norms for English and it might not disturb users if it is disallowed.

(13) On Monday a meeting with Jim Smith book.

The primary motivation for constructing the GD, however, is the close correspondence between grammatical constituents and elements of the CD. The GD thus serves as an aid to interpretation. Moreover, we need a syntactic level of representation to take care of strictly syntactic restrictions on phenomena such as reflexivization and long-distance dependencies.

It must be noted though that the interest in grammatical descriptions is not an interest in the structural potential of constructions, but with the structure appropriate for the corresponding content description on a particular occasion of use. While the grammar taken in isolation may allow several different GDs of a given input, the GD for a particular utterance is constructed in parallel with the CD using the other knowledge bases as well.

As said above an LFG-style formalism for the linguistic part of the description can be used, where the constraints on DODs that words and constructions are associated with can be formulated in the same way as functional constraints in LFG.[3] The GD and the CD are constructed incrementally and in tandem using a chart-parser for recognition of syntactic constituents.

To find the contextually appropriate interpretations and reduce the combinatorial explosion of alternative parses the parser is interacting with other processors that I call object

---

[2] Interestingly, Swedish is different on this point. *Avboka det på måndag* could mean either "Cancel it on Monday" or "Cancel that (= the one) on Monday".

[3] Cf. the use of situational schemata in Fenstad et al. (1986) In the illustrations below I use no f-structure level at all. Functional information is instead incorporated at the c-structure level. I do this here for the sake of brevity only and no theoretical claims are being made.

modules. Their purpose is to link DODs with discourse objects and evaluate the information in DODs against existing expectations. When a constituent is syntactically complete (or potentially complete) control is given to an object module which seeks to establish an object that is described by the DOD derived by the syntactic parser. Such a scheme should be based on a theory about the correspondence between syntactic structure and discourse object relations. The closer the correspondence the better it would be, but we definitely do not have an isomorphic correspondence. It seems, however, that the correspondences obey locality conditions of the kind that can be specified in the basic schemata of the LFG-formalism, the following being the most common ones:

Embedding:	$\uparrow = \downarrow$
Isomorphy:	$(\uparrow \text{Attr}) = \downarrow$
Discrimination:	$(\uparrow \text{Attr}) = \text{'Value}$
Percolation:	$(\uparrow \text{Attr}) = (\downarrow \text{Attr})$
	$(\uparrow \text{Attr2}) = (\uparrow \text{Attr1 Attr2})$

Similarly, we need a theory for the possible relations between lexical categories and constituent structure on the one hand, and for the relation between lexical items and DODs on the other. The relation between lexical heads and major syntactic constituents is in LFG spelled out as a condition that any f-structure must contain a semantic form as the value of the attribute PRED in order to be coherent and complete (Kaplan&Bresnan, 1982: 211f), where PRED-attributes primarily go with nouns, verbs and adjectives. In the present framework a similar correspondence can be stated in terms of DODs and the attribute Instance-of. However, we should allow Instance-of-descriptors to be associated with more than one word of a constituent as long as they have compatible values. This should be the case for expressions such as *Mr. Jim Smith*, where all words specify different attributes of a person, and for an adjective such as *dry* in (5) when it applies to wines.

I regard grammar rules as defining the internal composition of significant syntactic objects. By 'significant' is then meant significant for determining object descriptors. This means that I favour isomorphy and embedding as the local structural correspondences between GDs and CDs. The internal composition usually specifies one or more positions for lexical heads and other distinguished markers for that type of constituent. Rules for declarative sentences and NPs (which hold good for both Swedish and English) are shown below. VCOMP and NCOMP are variables over regular expressions of complements that are assigned variables from the lexical head.

R1: U —> { S[Decl] / S[Imp] / ... }

R2: S[Decl] —> NP[Subj] V[Fin] VCOMP SADJ*

R3: NP —> {DET/NP[Poss]} AP* N NCOMP REL*

As soon as a lexical head (or other marker) for a syntactic constituent has been recognized, such a constituent as well as a corresponding DOD can be postulated, the latter taking descriptors from both lexical head and structure. Associated with the rule that introduces declarative clauses we would have schemata such as:

DS1: $(\uparrow \text{Instance-of}) = \text{'Assertion}$
$(\uparrow \text{Agent}) = \text{<Score-board Speaker>}$
$(\uparrow \text{Recipient}) = \text{<Score-board Addressee>}$
$(\uparrow \text{Event}) = \downarrow$

A lexical entry for a word gives for each one of its different uses a syntactic category, a morphological sub-category (omitted here), a set of descriptive schemata and a structure of possible complements with associated descriptive schemata. The verb *cancel* has as one of its entries:

cancel; V; $(\uparrow \text{Instance-of}) = \text{'Cancel}$
NP[Subj]; $(\uparrow \text{Agent}) = \downarrow$
VCOMP: NP; $(\uparrow \text{Object}) = \downarrow$
PP; $(\uparrow \text{Time}) = \downarrow$

From DODs to Discourse Objects

The linguistic information can not give us a discourse object. Instead we need special modules that attempt to link DODs to discourse objects. There are different types of relations between DODs and discourse objects, however. Certain DODs should be linked to existing discourse objects (anaphoric pronouns, Proper Nouns), others should be used to constitute a discourse object (main declarative clauses, indefinite NPs in certain positions) and still others should be linked to a discourse object only indirectly (NPs and APs in predicative positions). Such information is also associated with words and constructions and we may encode it by special-purpose descriptors.

Suppose information concerning discourse status is encoded by means of an attribute Status with values such as Topical, Speaker, Addressee. An NP containing a definite article or the pronoun *it* is assigned such a descriptor from lexical entries of the following sort:

the; DET; { (↑ Status)=Topical
            / (↑ Status)='Known }

it; . { NP; (↑ Status)=Topical
            (↑ Sex)='Neutral / NP[it]; —}

If a DOD has the descriptor [Status: Topical] a module is activated which attempts to unify the given DOD (minus the Status-descriptor) with the DODs of the objects in the score-board field for topical objects. If this succeeds for exactly one of the topical objects, that object is chosen as the object picked out by the given DOD. We mark this on the DOD by assigning that object (i.e. its label) as value of a special attribute, say **Picks**. When the DOD is thus completed control is given back to the syntactic parser.

In the case of (4) such a matching would fail. Parsing can still continue with an alternative analysis of *it* as, say a purely formal element without links to a discourse object.

An object module may also be called to resolve structural ambiguities. In a parsing of (6) the syntactic processing would reach a state in which an ambiguity cannot be resolved on syntactic grounds. Let us assume the following rules and lexical entries in addition to those already stated.

R4: PP[p] —> P[p] NP
                    ↑=↓

R5: SADJ = { PP[on] / ... }
                ↑=↓

meeting; N; (↑ **Instance-of**) = 'Meeting
            NCOMP: PP[with];
                ↓ ∈ (↑ **Participants**)

            PP; (↑ **Time**) = ↓

Thus, the DOD associated with the PP *on Monday* can be consumed either by the DOD describing a topical meeting or the DOD describing the cancellation. If we match grammatically obtained DODs at every possible point of completion we would give control to the score-board processor as soon as we have found the phrase *the meeting* ignoring potential complements. The DOD would then be:

Instance-of:	'Meeting
Status:	Topical

If there is only one topical meeting, this match would succeed and we could then complete the constituent and attach it under the declarative S. This would also mean that NCOMP is set to NIL and that the PP will be

consumed by the verb. If there is no unique match in the score-board at this point, control is again given to the parser which looks for a PP-complement to the noun. It will find one, include its DOD in the meeting-DOD and again give control to the score-board processor. If there is now a unique match, parsing and interpretation will be completed succesfully; otherwise it will fail.

CONCLUSIONS

If we believe that users of NLIs think in terms of "doing things to things" and want to talk about those things in the same way as in ordinary language, e.g., by using pronouns and ellipsis, the NLI itself should be able to "think" in terms of things and understand when they are being talked about and how their saliency influence interpretation. Thus, an internal object-oriented representation language is suitable and a parser/interpreter that can make use of some knowledge about current discourse objects a necessity. As for the methods sketched briefly in this paper further work will be needed to determine whether they are adequate for their task.

ACKNOWLEDGEMENTS

REFERENCES

Brady, Michael and Berwick, Robert C. (1984): *Computational Models of Discourse*. Second printing. The MIT Press.

Dahlbäck, Nils and Jönsson, Arne (1986): *A System for Studying Human-Computer Dialogues in Natural Language*. Research report LITH-IDA-R-86-42, Linköping University, Department of Computer and Information Science.

Fenstad, Jens Erik, Halvorsen, Per-Kristian, Langholm, Tore and van Benthem, Johan (1986): *Equations, Schemata and Situations: A framework for linguistic semantics*. CSLI and Xerox Palo Alto Research Center.

Hayes, Philip J. (1984): *Entity-Oriented Parsing*. Department of Computer Science, Carnegie-Mellon University. Also in *10th International Conference on Computational Linguistics*, Stanford, 1984, pp. 212-217.

Kaplan, R. & Bresnan, J. (1982): *Lexical-Functional Grammar: A Formal System for Grammatical Representation*. In Bresnan (ed.) (1982) *The Mental Representation of Grammatical Relations*. The MIT Press, Cambridge, Mass. pp. 173-281.

Lewis, David (1979): *Scorekeeping in a Language Game*. In R. Bäuerle, U. Egli and A. von Stechow (eds.): *Semantics from Different Points of View*. Springer-Verlag, 1979: 172-187.

Sidner, Candace L. (1984): *Focusing in the comprehension of definite anaphora*. In Brady&Berwick pp. 267-330.

Tomita, Masaru, and Carbonell, Jaime G. (1986): *Another Stride Towards Knowledge-Based Machine Translation*. Proceedings of COLING '86, University of Bonn, pp. 633-38.

Webber, Bonnie L. (1984): *So what can we talk about now?* In Brady&Berwick pp. 331-371.

Wilks, Yorick, Huang, Xiuming & Fass, Dan (1985): *Syntax, Preference and Right Attachment*. In *Proceedings of the Ninth International Joint Conference of Artificial Intelligence*, Los Angeles, 1985, pp. 779-784.