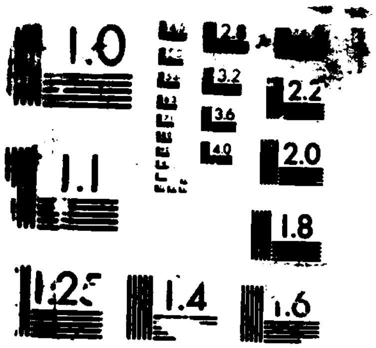
AD-A189 201 THESIS PROPOSAL: A PLAN-BASED APPROACH TO CONVERSATIONAL IMPLICATURE(U) ROCHESTER UNIV NY DEPT OF COMPUTER SCIENCE E HINKELMAN 30 JUN 87 TR-202 1/1

UNCLASSIFIED N00014-80-C-0197 F/G 5/7 NL



MICROCOPY RESOLUTION TEST CHART

Thesis Proposal: A Plan-Based Approach to Conversational Implicature

Elizabeth Hinkelman
Computer Science Department
The University of Rochester
Rochester, New York 14627

TR 202
June 30, 1987

rochester

Department of Computer Science
University of Rochester
Rochester, New York 14627

Thesis Proposal: A Plan-Based Approach to Conversational Implicature

Elizabeth Hinkelman
Computer Science Department
The University of Rochester
Rochester, New York 14627

TR 202
June 30, 1987

Implicatures are those propositions that must be inferred from an utterance in order to make sense of it in context, but which are not truth conditions or entailments of that utterance. Conversational implicatures are those that can be explained by general principles rather than lexical choice. Conversational implicature was defined in philosophy and discussed in linguistics, but lacks an adequate account in any discipline. For the computational linguist, this means that much information conveyed indirectly in discourse has no computational model.

We show that knowledge of goals and plans is necessary for the computation of some implicatures, and very useful for a much larger class of implicatures. Our model uses a set of inference rules about STRIPS-style plans for implicature computation. It incorporates a computational model of speech acts (based on propositional attitudes) and surface speech acts (based on linguistic features). We propose an implementation, and relate it to other research in computational linguistics.

This work was supported in part by NSF research grant no. IST-8504726, ONR research contract no. N00014-80-C-0197, NIH Public Health Service research grant no. 1 R01 NS22407-01, and an Eastman Kodak Company Fellowship. We thank the Xerox Corporation University Grants Program for providing equipment used in the preparation of this paper.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
TR 202

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER
AD-A189 201

4. TITLE (and Subtitle)
Thesis Proposal: A Plan-Based Approach to Conversational Implicatures

5. TYPE OF REPORT & PERIOD COVERED
Technical Report

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Elizabeth Hinkelman

8. CONTRACT OR GRANT NUMBER(s)
N00014-80-C-0197

9. PERFORMING ORGANIZATION NAME AND ADDRESS
University of Rochester
Computer Science Department
Rochester, NY 14627

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Defense Advanced Research Project Agency
1400 Wilson Blvd.
Arlington, VA 22209

12. REPORT DATE
June 30, 1987

13. NUMBER OF PAGES
31

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
Office of Naval Research
Information Systems
Arlington, VA 22217

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Distribution of this document is unlimited

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

18. SUPPLEMENTARY NOTES
None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
implicature, linguistics, inference rules, computational model of speech acts

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
Implicatures are those propositions that must be inferred from an utterance in order to make sense of it in context, but which are not truth conditions or entailments of that utterance. Conversational implicatures are those that can be explained by general principles rather than lexical choice. Conversational implicature was defined in philosophy and discussed in linguistics, but lacks an adequate account in any discipline. For the computational linguist, this means that much information conveyed indirectly in discourse has no computational model. ~~We show that~~ knowledge of goals and plans is →

DD FORM 1473, 1 JAN 73 — EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Continued)

necessary for the computation of some implicatures, and very useful for a much larger class of implicatures. Our model uses a set of inference rules about STRIPS-style plans for implicature computation. It incorporates a computational model of speech acts (based on propositional attitudes) and surface speech acts (based on linguistic features). We propose an implementation and relate it to other research in computational linguistics.

Table of Contents

1. Conversational Implicature

1.1. Introduction

You are walking down a road when you see a car on the shoulder. A person approaches you with a gas can in hand, and says,

(1) a: I'm out of gas.

You reply,

b: There's a gas station around the corner

The person is now entitled to conclude that as far as you know, the gas station is open, will actually sell the gas, and so on. These conclusions are not strict logical entailments of the reply nor its truth conditions. Rather, they follow from the sentence in context and general knowledge of human communication and behavior. To see that the conclusions really can be drawn, consider the person who after going around the corner finds you again and says

c: They were closed.

If you say,

d: I know,

the person becomes aware that you are acting uncooperative or somehow very ignorant. If you say,

e: Oh, sorry. I guess it's closed on Sundays.

(and it's Sunday), the person must conclude that you had forgotten this.

This paper proposes a computational model that draws the (gas-seeking) agent's conclusions from an exchange. The class of conclusions addressed is a subset of those referred to as "conversational implicatures" by the philosopher H. Paul Grice [Grice 75], from whose work this example was adapted. The general approach combines discourse processing techniques and plan recognition methods to infer the connections among utterances, and a set of constraints on their implicatures.

1.2. The Philosophy Literature

The topic of conversational implicature falls under the heading of pragmatics, which concerns the use and appropriateness of utterances in context as opposed to their context-independent meaning (semantics) or syntactic structure (syntax.) Related pragmatic issues include presupposition, which under some definitions overlaps with implicature but also has semantic variants, and speech act theory, probably the best known topic in pragmatics.

In defining conversational implicature, Grice was attempting to isolate an aspect of language which could be explained on general behavioral principles rather than the traditional truth-functional view that obtains in formal logic. Austin [Austin 62] had already pointed out that some sentences constitute events themselves (as opposed to merely describing events). His *performatives* include such forms as

(2) A: I christen thee the Queen Mary.
B: I now pronounce you man and wife.

Note that the word "hereby" can be inserted into such sentences, emphasizing their role in achieving some social condition. To say anything is to perform an action, but performative statements have particularly explicit effects. Searle [Searle 75] extended the theory of speech acts to describe utterances such as

(3) A: Could you please pass the salt?
B: Do you have the time?

in which the literal request differs from the actual use of the utterance. These *indirect* speech acts were explained in terms of commonsense reasoning about the utterer's intentions, using context to point to an adequate interpretation of the utterance. The theories of speech acts and implicature are both concerned with the roles of agents' beliefs and intentions in language use, and with language use as a subclass of human behavior.

David Lewis [Lewis 79] described a behavior that he called "accommodation," by which people adjust their understanding of a situation to make new developments fit. Thus when someone at an informal evening gathering offers to make restaurant reservations, that person becomes the leader by accommodation on the part of each person present. A linguistic example would be, if we find we've mistaken what someone said, we may apologize, but we act for the rest of the conversation as if we had understood the first time. Thomason [Thomason 84] pointed out that conversational implicatures appear to function similarly: the implicature in 1b (that the speaker thinks that the gas station is likely to be open) must be added to the context to make 1b be a reasonable response to 1a. A good computational account of conversational implicature might have something to say about the nature of accommodation. McCafferty [McCafferty 86] proposed to borrow plan structures from Artificial Intelligence as a basis for understanding conversational implicature. His work will be described later.

1.3. The Gricean Framework

Grice himself proposed a taxonomy of communication (Fig. 1) in which conversational implicature has an especially problematic role. Grice distinguished among the utterance (what is said), its meaning (propositional content when taken in isolation, or sentence meaning), and its implicatures. Implicatures are additional conclusions that can or must be drawn (based on the

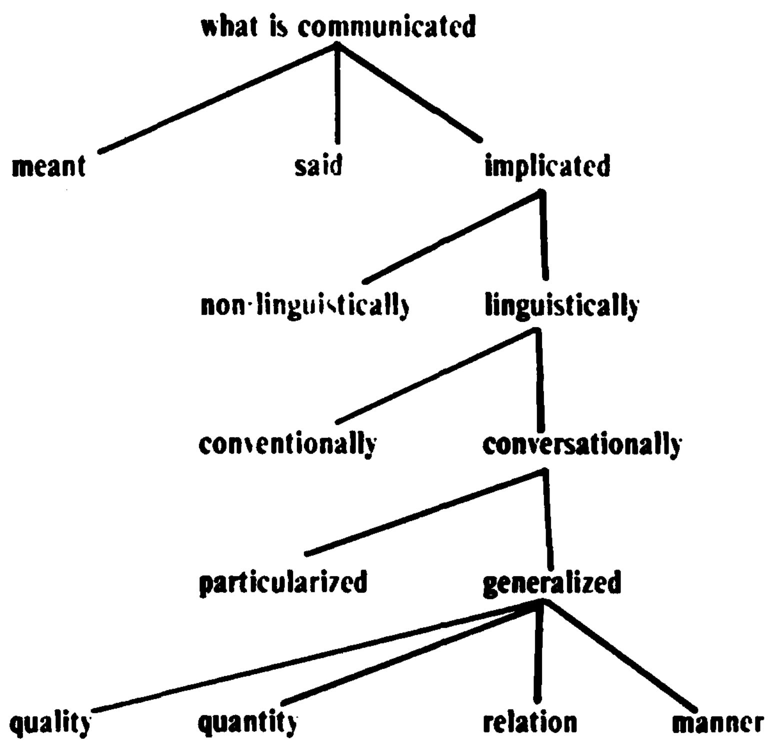


Fig 1: Grice's Taxonomy

utterance, a "normal" body of world knowledge, and the context) for the utterance to make sense in that context. They are characterized as cancellable, that is, they can be denied without denying the meaning of the utterance. For instance, you could say "There's a gas station around the corner, but I think it might be closed." This distinguishes implicatures from implications. Implicatures may be linguistic or non-linguistic[1], and then conventional or conversational.

Conventional implicatures are those which can be attributed to the use of a particular word or phrase. The use of "could" as a request in "Could you pass me the salt?" is a convention. Grice also said that the semantic relationship asserted by "therefore" in "He is an Englishman, he is, therefore, brave" is a conventional implicature, although its cancellability is implausible. In fact words and phrases may serve many roles in addition to contributing to the propositional content of an utterance; we shall return to this point later.

---

[1]A non-linguistic analogue of conventional implicature might be occasioned by failing to bow to a person of higher rank. The lower-ranked person might cancel the insult, saying "Your lordship will pardon his humble servant's injured back." A "conversational" implicature might be the conclusion that, when someone hands you a flute, that you are expected to play it. One would expect this to work in any culture, though not in every context. Another "conversational" implicature is the basis for an old comic skit: A man leaves the doctor's office wearing a neck brace that forces his head back, but the brace is covered by his coat. People on the street notice his staring upward and start doing the same, until everyone around is gawking at nothing. No one cancels the implicature that there is something interesting to look at. I owe this example to David Sher

Conversational implicature is thus what remains in the category of linguistically effected implicatures when those that cannot be "detached" from conventions of word or phrase use have been removed. They are strongly affected by context. So far, the issues to contend with in accounting for conversational implicatures are:

- Cancellability. Implicatures non-monotonic, since they can be cancelled. but the propositional content of the sentence is unaffected.
- Context effects. Suppose that in our initial example the gas station is several miles down the road, just on the horizon. The person will become chagrined rounding the corner, since as it turns out you have also implicated that the station is easily reachable. If the person had been biking rather than walking, it might not be so bad. So there is a secondary implicature about the distance of the gas station that depends on your beliefs about the form of transport involved[2]. So we have to represent and utilize context, and beliefs about it.
- Detachability. Conversational implicatures must be explained in terms of general principles or systems, as they can be detached from particular words or phrases.
- Indeterminacy. There may be several possible implicatures or an open list of them in some situations, and a proper computational account must identify these cases when people can.
- What mechanism can we use to find any implicatures at all for a particular utterance and context?
- Which information is relevant in computing implicatures of a given utterance?

Grice's approach to conversational implicature, being philosophical rather than computational, consists of a list of principles of discourse (Maxims, Fig. 2) and a description of their use. There have been many subsequent attempts to recast and reformulate their operation. and a good computational account of implicature would shed some light on this endeavor. Grice's original proposal is described here, followed by a summary of subsequent work.

Cooperativity is the overarching ideal that, at the discourse level at least, conversants agree on a goal of the conversation and speak accordingly. The maxims are more specific, and the cooperativity assumption allows a hearer to infer that speakers are obeying the maxims even when they appear not to. Prima facie maxim violations can be redeemed (or accommodated) by making certain assumptions, namely conversational implicatures. Thus in (4)

(4) A: Where did the eggs go?
    B: John ate some of them.

it is implicated that the speaker doesn't know what happened to the rest of the eggs, since she is

[2]This is based on a constraint on the Goto step of the plan. See below

Quantity:

(1) Make your contribution as informative as is required (for the current purposes of the exchange).

(2) Do not make your contribution more informative than is required.

Quality: Try to make your contribution one that is true.

(1) Do not say what you know to be false

(2) Do not say what you lack adequate evidence for.

Relation: Be relevant.

Manner: Be perspicuous.

(1) Avoid obscurity of expression.

(2) Avoid ambiguity.

(3) Be brief (avoid unnecessary prolixity).

(4) Be orderly.

Fig 2: Grice's Maxims

assumed to be cooperative but has failed to answer for all of the eggs. Submaxim Quantity 1 requires the speaker to answer the whole question if she can.[3]

An example of Quantity 2 is

(5) John ate some of the eggs. but that's not the whole story.

Here, since the speaker has elaborated on the obvious (that some isn't all) the implicature is that what happened to the rest of the eggs is particularly interesting. The submaxims of Quality are relatively straightforward. but Relation is both broad and mysterious. Relation is the principle operating in the gas station example: 1b, to be taken as a relevant response to 1a, is best regarded as a suggested solution to the problem. The utterance would be pointless if the gas station were thought to be closed. or out of gas. or unreachable. etc. But it remains to say how to identify and use relevance.

The submaxims of Manner are as much about language as about rational behavior. If instead of 1b one said

[3]Note that although all of the maxims are intended to be relativized to the "purposes of the exchange," these purposes may never have been explicitly stated.

(6) There's a building with gas pumps around the corner.

one would implicate that the building is not a gas station (Manner 1). If someone is ambiguous

(7) A: Who went to the store?
A: One of the boys.

they implicate that they either don't know or don't think it matters (Manner 2). Saying something the long way is significant too (Manner 3).

(8) A: He told tales of his youth.
B: He uttered a series of sentences which, when taken together in groups, could be considered anecdotal of his younger days.

In this case it suggests that the storyteller was rambling and longwinded. So far we've advocated merely using the best expressions at hand, but consider the following:

(9) A: He cleaned up, cooked, and ate
B: He cooked, cleaned up, and ate.
C: He cooked, ate, and cleaned up.

The same words have different implications about the ordering of events, when used in a different order (Manner 4). Grice's use of the word "orderly" for this phenomenon may seem to be a cheap shot until we note that this can be made to work for spatial and conceptual contiguities as well.

A final aspect of conversational implicature is that the above maxims apply to what Grice called *generalized* conversational implicature. He claimed that there was also *particularized* conversational implicature, as when

(10) Jones is seeing a woman this evening.

suggests that the woman is not his spouse, sister, or close platonic friend. The idea is ostensibly that although you could substitute semantic equivalents for "woman" to the same effect, no general behavioral principle applies. The claim that "seeing a woman" is not a case of Manner 1, 2 or 3 seems dubious, although they would need to be augmented by a theory of salient information.

1.4. The Implicature Literature

Some discourse phenomena can be handled by extending traditional lexical and syntactic approaches. Weischedel integrated computation of presupposition and entailment into the parsing process [Weischedel 79]. He associated rules with his syntactic categories, for instance, the assertion that noun phrases have a referent. Likewise Karttunen and Peters provided an account of conventional implicature using Montague grammar [Karttunen 79]. They treated implicature-bearing words as special functions, as quantifiers are treated in conventional Montague grammar.

Conversational implicature, however, is by definition a phenomenon that cannot be pinned to particular words but depends on more general relationships between utterances and contexts. The most profitable line of conversational implicature study to date has concerned itself with Quantity.

Horn [Horn 72] asked in his linguistics dissertation, "When do presuppositions bear suspenders?" Suspension of presuppositions can here be understood as cancellation of implicatures. Horn observed that certain sets of words can be ordered on the basis of informativeness. His *scales* include the integers, temperature, golf scores, time, the logical quantifiers and operators, and many more. He proposed rules to determine when alternate values on a scale are implicated or not, depending on the asserted (or denied) value. (4) is an example of an utterance with implicatures based on a rich scale of quantifiers. The definitive treatment to date is that of Hirschberg [Hirschberg 85]. She generalized scales to partially ordered sets, revised Horn's axioms about them, and gave a formal computational account of the phenomenon. She included a rule for computing scalar implicatures when the scales understood by speaker and hearer differ. She suggested that identification of relevant scales, a significant problem in its own right, could be done using intonation clues. The plan-based approach (to be described) has potential for identifying relevant scales as well as for improving the definition of scales (consider ordering the set of gas stations that you know of, based on distance, prices, and service.)

Much of the remaining work on conversational implicature took the form of an analysis and extension of Grice's Maxims. Harnish [Harnish 76] added his Principle of Charity: construe an utterance so as to violate as few or as unimportant Gricean Maxims as possible, ordering the Maxims. This work could be understood as limiting somewhat the search for implicatures. He also was responsible for subdividing Manner 4 into time and "space," and a number of other observations. Gazdar [Gazdar 79] did a very painstaking study of implicature and presupposition that contributes primarily a distinction between potential and actual implicatures. A construct may license several potential implicatures, some of which may be blocked from becoming actual implicatures. (if they coincide with the propositional content of the sentence, for example.) Horn [Horn 84] reclassified conversational implicatures into two categories based on minimizing speaker and hearer effort respectively. This distinction is useful in that it points out what work can be assumed to be done by the speaker and what by the hearer. The distinction and its associated dialectic are terms which can be usefully applied to many linguistic phenomena.

Sperber and Wilson [Sperber 86] discuss relevance at length in their book by that title, arguing that Grice's maxims were better discarded in favor of comparative evidential reasoning. Their point is that models which are qualitatively similar to their phenomena can be more illuminating than black-box models. And it's certainly true that a stochastic simulation of particles moving in a gas is informative in a way that the higher-level gas laws are not. Sperber and Wilson neglect that the converse is also true: a summary of the rules that a system satisfies says more about the global system state than raw data would. They neglect that no one, including Grice, pretends that his rules can be implemented directly. And they neglect to show that comparative evidential reasoning, applied in any way, produces intuitive results. Since evidential reasoning is a major open problem in Artificial Intelligence, it is unlikely that Sperber and Wilson will solve it singlehanded. Emphasis on access time is a mainstay of the knowledge representation literature to which they have added little leverage.

My methodology is as follows. Given some examples, I give rules that describe the inferences made. Then I provide one computational system that implements these rules for testing, remaining where possible consistent with psychological findings. Gibbs [Gibbs, Jr. 84], for instance, shows that people don't compute literal meanings before figurative ones in understanding sentences, making certain heavily serialized algorithms implausible for sentence understanding. Then where it is called for, one could attempt to introduce simple evidential reasoning techniques to refine the model. For the most part psychology has had little to say about implicature specifically or about possible mechanisms for related phenomena, though.

2. The Plan-Based Approach

2.1. Plans and Domain Plans

The central claim of this proposal is that an analysis of intention is essential to computing some conversational implicatures, and a great help in finding many others. Consider arriving at the trailhead with some hiking friends, and while people are lacing their boots, one asks you,

(11) *Do you have a watch on?*

You answer the question, concluding that the person thinks she may later want to know the time. But suppose you were sitting in a boring lecture when someone pencilled the same question on your pad. In this case she likely does want to know the time then, so you tell her. The question is not a conventional request for the time, but you have recognized the intent. Without an analysis of intention, the two cases would be indistinguishable.

Ambiguity about plans can lead to conversational difficulties. A student who was thinking about buying some candy went to the library desk.

(12) i: Can I have change? (proffers $20.)
ii: Not for a twenty.

The student looked for a friend, who exchanged two tens, but on his return was told that in fact change was available only for the photocopy machines. The librarian had licensed the implicature that the student could have change for smaller bills, and that the student's plan would work, on the assumption that the plan was to make copies. The student had computed the implicatures that he could have change for a smaller bill and that his plan would work, on the assumption that the plan was to buy candy. Under both interpretations the literal content of the librarian's statement is true, but the implicatures are different. Had the librarian correctly recognized the student's plan, he would have been obliged to state that he was unable to change any denomination.

Disjunction can yield implicatures, depending on the plan. If someone in your family asks you where the window screens are, and you say

(13) They're in the basement or the attic.

you implicate that you don't know which place the screens are in. (This can be explained in terms of Quantity, Relation, or Manner.) But consider supervising a treasure hunt for some children. When they come back begging for a hint, you make the same reply. They don't conclude from your hint that you don't know which. (One might use Gazdar's terms to explain this: the potential implicature that you don't know is cancelled by the fact that you do, or by the plan for the game.) The goal of the utterance changes its implicatures.

Or take the gas station example once more. This time you're in a conservative town like Montreal or Geneva:

(14) i: I'm out of gas.
ii: Too bad, it's a holiday.

The driver is out of luck. But how did your reply make sense? It was based on the idea that the driver needs gas, and the way to get it is to buy it at a gas station, but gas stations are closed on holidays. If there were no shared knowledge of plans and the world, the exchange would be unintelligible.

The early literature on plans and planning, exemplified here by such works as [Fikes 71, Nilsson 80, Sacerdoti 74, Sacerdoti 80] is well known to students of Artificial Intelligence. In general a plan instance is a data structure representing an intended course of action on the part of an agent. We will also use it as a shorthand for the planner's beliefs and intentions with respect to this course of action. It may have a name, and in some theories it is uniquely determined by its agent and time. This view of plans is overly restrictive; in particular it will need to be supplemented by a theory of multiagent plans. However, this work remains to be undertaken. Grosz and Sidner have begun to consider these issues [Grosz 87].

A plan schema defines a plan type, which is part of a type hierarchy. A plan schema is written with a header that specifies the type name and parameters of the schema. We will use the predicate PARAMETER(term, plan) to indicate that the term unifies with some parameter of the plan. Plan parameters must include an agent and a time of execution, which will differ in general from the time of planning. PRECOND(proposition, plan) asserts that the proposition must hold in order for the plan to be executed successfully. CONSTRAINT(proposition, plan) is similar, except that it is used to denote a condition which an agent cannot plan to achieve, whereas preconditions are intended to be achievable[4]. EFFECT(proposition, plan) denotes a state resulting

[4]Roughly speaking, this distinction is embodied in Pelavin's CHOOSIBLE predicate [Pelavin 86], defined to hold when, in every possible future, the chooseable state does not exist but can be made to exist by executing some plan. My only quibble is that this is a

header: Get-Gas(agent, loc, time, gas)
preconds: AT(gas, loc, time)
decomp: Goto(agent, loc, time)
Get(agent, gas, time)
effects: HAS(agent, gas)

header: Buy-Gas(agent, seller, loc, time, gas)
preconds: OWN(agent, price(gas))
constraints: OWN(seller, gas) ∧ AT(seller, loc, time) ∧ AT(gas, loc, time)
decomp: Goto(agent, loc, time)
Give(agent, seller, price(gas), time)
Give(seller, agent, gas, time)
effects: OWN(seller, price(gas)) ∧ OWN(agent, gas) ∧
~OWN(agent, price(gas)) ∧ ~OWN(seller, gas).

header: Steal-Gas(agent, owner, loc, time, gas)
preconds:
constraints: ~AT(owner, loc, time) ∧ AT(gas, loc, time) ∧ ~ALARMED(gas)
decomp: Goto(agent, loc, time)
Siphon(agent, gas, time)
Getaway(agent, time)
effects: HAS(agent, gas) ∧ CROOK(agent)

Fig 3: Plan Hierarchy

from successful execution of the plan. DECOMP(actions, plan) specifies that a list of actions (which may themselves be plans with decompositions, or primitive actions with none) performed in order (which should be specified in a good temporal logic) constitutes an execution of the plan. One plan may have several decompositions. STEP(action, plan) asserts that action occurs in some decomposition of plan. This last predicate is supplied for convenience only; the others taken together suffice to define a plan type. Another convenience we will make use of is to denote the set of a given plan's parameters as α (for "arguments"), constraints κ, preconditions π, and so on.

An agent who can be said to have a plan has certain beliefs with respect to the plan. These beliefs were spelled out by Pollack for purposes of plan recognition in question-answering, where plans may be erroneous. Rather than present her formalism here, I'll sketch in English the beliefs that seem to be useful for conversational implicature. To have a plan at a time t1, to be performed at a time t2, the agent must believe

- that the plan's constraints will hold (or are likely to hold) at t2,

---

little too strong: the only way to ignore some unlikely world is not to represent it in the data base. Further, we might want to count worlds in which the state obtains but its negation is achievable by planning. But these are much harder to formalize.

- that the plan's preconditions can be achieved by t2 and that the agent intends to achieve them,
- that each of the actions in the decomposition is performable at t2 (this is a naive idea of time),
- that each has some useful role in the plan, and that the agent actually intends to do them at t2,
- and that the effects of the plan will hold after t2, which would not be the case were the plan not executed. The agent may in fact only intend some of the effects, others of which might be considered undesirable side effects. Pollack's formalism also provides for explanations of the actions, which we don't need for our purposes. Note that as stated the agent can't have a plan involving any other agent, since the agent actually intends to perform all the actions. Pollack's presentation of plan structures is foundational for mine, since it is the most careful and well-developed to date[5].

There are three major processes which are based on plan structures. Planning combines familiar actions into novel plans for novel situations. We will avoid the issues involved in planning, for the sake of focus. Plan execution takes an existing plan and performs its steps. A longstanding issue here is feedback: note that it is central to an understanding of dialogue. Plan recognition is the process of identifying an agent's intentions and actions. This too, as we shall see, is integral to understanding dialogue.

A simple domain plan hierarchy appears in Fig. 3. The broad arrows connecting both the Buy-Gas and Steal-Gas templates with the Get-Gas template denote that Get-Gas is an abstraction from the other two: anything that fits the description of a Buy-Gas or Steal-Gas action will also fit the description of Get-Gas. (This is Henry Kautz's definition of abstraction.) Thus when there is not enough information to identify a plan to acquire gas unambiguously, Get-Gas allows a certain amount of reasoning about the plan to proceed.

2.2. Plan-Based Discourse

The plan-based approach to discourse makes use of plans not only as an organization of the subject matter of discourse but as a description of discourse itself [Allen 83, Appelt 85, Litman 85]. In other words, not only do we talk about plans, but our talking can be represented as plans and actions. This permits a uniform mechanism for modelling discourse along with other intentional behavior. Allen first used this technique for implementing speech act theory, while Appelt extended the approach to generate utterances that satisfy as many discourse goals as possible. In effect, then, speech acts can be reduced to the

[5] I will, however, avoid any claims associated with her view of "plans as mental phenomena" since it suggests a psychological account that none of us, including Pollack, are prepared to offer

header: Introduce-Plan(speaker, hearer, action, plan, time)
decomp: Request(speaker, hearer, action, time)
constr: STEP(action, plan) ∧ AGENT(action, hearer)
effect: WANT(hearer, plan) ∧ NEXT(action, plan)

header: Ask(speaker, hearer, var, prop, time)
decomp1: Request(speaker, hearer, Informif(hearer, speaker, prop, time), time)
Informif(hearer, speaker, prop, time)
decomp2: Request(speaker, hearer, Informref(hearer, speaker, var, prop, time), time)
Informref(hearer, speaker, var, prop, time)

header: Request(speaker, hearer, action, time)
decomp1: Surface-Request(speaker, hearer, action, time)
decomp2: Surface-Request(speaker, hearer,
INFORMIF(hearer, speaker, CANDO(hearer, action), time), time)
decomp3: Surface-Inform(speaker, hearer, ~CANDO(speaker, action), time)
decomp4: Surface-Inform(speaker, hearer, WANT(speaker, action), time)
decomp5: Surface-NP(term, time) -- with condition PARAMETER(term.action)
effect: BEL(hearer, WANT(speaker, PERFORM(hearer, action)))

header: Inform(speaker, hearer, prop, time)
decomp1: Surface-Inform(speaker, hearer, prop, time)
constr: KNOW(speaker, prop)
effect: KNOW(hearer, BEL(speaker, prop))

header: Informif(speaker, hearer, prop, time)
decomp1: achieve KNOW(hearer, BEL(speaker, prop))
decomp2: achieve KNOW(hearer, BEL(speaker, ~prop))
constr: KNOWIF(speaker, prop)
effect: KNOWIF(hearer, BEL(speaker, prop))

header: Informref(speaker, hearer, x, P(x), time)
decomp1: achieve KNOW(hearer, BEL(speaker, P(A)))
constr: KNOWREF(speaker, x, P(x))
effect: KNOW(hearer, BEL(speaker, P(A)))

KNOW(s, p) ::= p BEL(s, p)
KNOWIF(s, p) ::= KNOW(s, p) ∨ KNOW(s, ~p)
KNOWREF(s, x, p(x)) ::= ∃x BEL(s, p(x))

Fig 4: Discourse Plans and Actions

beliefs and intentions that constitute a plan. Litman developed a theory of metaplans that can be used to restructure domain or discourse plans; these plans are useful for describing both discourse that leads to domain plan modification and modification of discourse plans when such phenomena as interruption and clarification subdialogues are encountered.

A subset of Litman's discourse plans and actions is shown in Fig. 4. The belief operator is written BEL, and KNOW is taken as true belief. The predicate KNOWREF allows us to distinguish constants that the hearer creates in parsing referring expressions from constants that have independent significance for the hearer. If the agent KNOWREFs an expression, she can identify its referents. KNOWIF asserts that the agent knows the truth value of a proposition. These two conditions are the results of their corresponding Inform actions in the so-called "brain surgery" theory of communication; I have written here instead merely that the hearer believes these things of the speaker, upon which the hearer may or may not choose to accept the information. The most elegant approach to the problem of representing what the hearer gets from the speaker is the default logic of Perrault [Perrault 87]. An Ask plan simply encodes a query and response of each type.

The bulk of the current proposal is concerned with finding the relationship between a response to the agent's query and the agent's plans. Introduce-Plan is an act, defined by Litman, in which a speaker informs a hearer indirectly of the plan, by asking the hearer to perform the next step in the plan. If we wish to regard "There's a gas station around the corner." as a suggestion of an

header: Surface-Inform(speaker, hearer, proposition, time)
effect: BEL(hearer, WANT(speaker, KNOW(hearer, proposition)))

corresponds to declarative sentences.

header: Surface-Request(speaker, hearer, action, time)
effect: BEL(hearer, WANT(speaker, PERFORM(hearer, action)))

Surface-Request(.... action) -- imperative sentences
Surface-Request(.... Informif(..)) -- yes/no questions
Surface-Request(.... Informref(..)) -- wh-questions

form	aux. inversion	subject	wh-term	subjunctive
declarative	-	+	-	+/-
y/n question	+	+	-	+/-
imperative	-	-/you	-	-
wh-question	+/-	+	+	+/-
noun phrase	0	-	-	0

Fig 5: Surface Speech Acts

entire plan rather than an Informref of a location variable, we need a speech act that allows introduction of an entire plan. One might have thought that Introduce-Plan would suffice, but it proved to have surface linguistic realizations that were unacceptable in the context of the gas station example. This led us to a complete reformulation of surface speech acts. Just as illocutionary speech acts consist in beliefs and intentions, we will construe surface speech acts as a logical form, a set of linguistic features, and any necessary beliefs and intentions. The main advantages of this method are that it is simple and readily extensible to accommodate more linguistic information, and that the features may be realized syntactically, lexically, phonetically, or intonationally. For convenience, often used surface speech acts will be given names.

The surface speech acts of prior theories are Surface-Inform, for declarative sentences, and Surface-Request, for imperatives and questions. Yes/no questions are Surface-Requests of an Informif, and wh-questions, Surface-Requests of an Informref. Fig. 5 shows their definitions. I have added a table of the surface features that characterize these utterance moods. The designation 0 is used to describe features which cannot be said to be present or not since they depend on another feature which is absent. A / indicates alternative possibilities. A noun phrase appearing alone seems to be an object rather than a subject.

Additionally, Fig. 6 shows the variety of utterances that can be obtained by varying a few key features: the subjunctive, conditional, and the word "please". It would be desirable also to include intonational features, which are currently an active area of study [Pierrehumbert 87]. This will require some care, since intonation can override the usual significance of both syntactic and lexical choices: "Would you PLEASE go to the store!" Further, there may be better ways to account for lexical items than by doubling the size of the feature space: by associating them with illocutionary speech acts, say. But such work remains to be undertaken.

The surface speech acts are then used to redefine the decompositions of the Request speech act, and to define a new Suggest act for the gas station sort of example. It will be necessary to reformulate Ask and Inform in these terms as well, but the Ask shown here is not revised. It would also be useful to characterize what makes a successful Informref and Informif, which may involve both semantic and linguistic information. A reasonable start on this would be to integrate mechanisms resembling Goodman's reference algorithm [Goodman 86].

The surface speech acts referred to by name are now as follows: Surface-Inform corresponds to nos. 0-3 of Fig. 6, Surface-Imperative 4-7, Surface-Simper 7 (and maybe 5), Surface-Request(...Informif) 8-15, Surface-Request(...Informref) 16-19, and Surface-NP 20-23. The Request plan now has several decompositions. They include the imperative form, queries about the parameters, preconditions and constraints of the action, statements about standard conditions of the action, and if the action requested is a speech act, there is a corresponding form of question for it: yes/no or wh.

Suggest is fairly straightforward, the important part being that suggestors don't say "please". There are two idiomatic forms for suggestions that might be mistaken for wh-questions, Howabout and Whatif. The others use a parameter, a step in the plan, or the plan itself to

subjun	if	please	reference number	examples
declarative				
+	+	-	0	If they'd listen... If you want, ...
+	-	-	1	You could go ...
-	+	-	2	If it rains, it pours. If you go 'round ...
-	-	-	3	It may rain.. There's a gas station...
imperative				
-	+	+	4	If you go to the store, please get me some bread.
-	+	-	5	If you go to the store, get me some bread.
-	-	+	6	Please get me some bread.
-	-	-	7	Get me some bread.
interrogative (y/n) (can/could will/would shall/should might/may must ought used need dare have be do)				
+	+	+	8	If you go to the store, could you please
-	+	+	9	If you go to the store, can you please
+	-	+	10	Could you please
-	-	+	11	Can you please
+	+	-	12	If you go to the store, could you get... If there were ghosts, would...
-	+	-	13	If you go to the store, can you get... If they're isolated, can birds...
+	-	-	14	Could you get ... Were you going? Could it rain?
-	-	-	15	Can you get... Is it raining? Do you know .. ?
interrogative (wh) (who what where when why how)				
+	+	-	16	If it rained, where would the worms go?
-	+	-	17	If it rains, where do the worms go?
+	-	-	18	Where could the worms be?
-	-	-	19	Where are the worms?
noun phrase				
	+	+	20	?If you go to the store, bread and milk please.
	+	-	21	?If you go to the store, bread and milk.
	-	+	22	Bread and milk, please.
	-	-	23	Bread and milk.

Fig 6: Surface Speech Acts: More Features

header: Request(speaker, hearer, action)
decomp1: Surface-Imperative(speaker, hearer, action)
decomp2: Surface-Inform(speaker, hearer, WANT(speaker,action))
decomp3: Surface-Inform(speaker, hearer, ~state), EFFECT(state,action)
decomp4: Surface-Request(speaker, hearer, Informif(hearer, speaker, P))
PRECOND(P, A) or CONSTR(P, A)
decomp5: Surface-Request(speaker, hearer, Informref(hearer, speaker, x, P(x)))
PARAM(x, action)
decomp6: Surface-Inform(speaker, hearer, ~P), PRECOND(P,A) OR CONSTR(P, A)
decomp7: Surface-NP(speaker, hearer, NP) PARAM(NP, action)
decomp8: Surface-Request(speaker, hearer, Informif(hearer, speaker, P))
action = Inform(hearer, speaker, P)
decomp9: Surface-Request(speaker, hearer, Informref(hearer, speaker, x, P(x)))
action = Inform(hearer, speaker, x, P(x))
decomp10: Surface-Request(speaker, hearer, Informplan(hearer, speaker, goal, plan))
action = Inform(hearer, speaker, plan)
decomp11: Surface-Inform(speaker, hearer, ~state), EFFECT(state,?A*Action)
action = Inform(hearer, speaker,?A*Action)
precond: WANT(speaker, action)
effect: BEL(hearer, WANT(speaker, PERFORM(hearer, action)))

header: Suggest(speaker, hearer, action)
decomp1: Howabout(speaker, hearer, action)
decomp2: Whatif(speaker, hearer, action)
decomp3: Informref(speaker, hearer, x, P(x)) PARAM(x, action)
decomp4: Surface-Inform(speaker, hearer, action1)
decomp5: S-Simper(speaker, hearer, action1)
precond: STEP(action1, action) or action1 = action
effect: BEL(hearer, CANDO (hearer, action))

header: Informplan(speaker, hearer, goal, plan)
decomp1: Suggest(speaker, hearer, plan)
constr: BEL(speaker, WANT(hearer, goal)) ∧ BEL(speaker, ACHIEVES(plan, goal))
effect: BEL(hearer, BEL(speaker, ACHIEVES(plan, goal))) ∧ Knowref(hearer, plan)

header: Ask(speaker, hearer, var, prop)
decomp1: Request(speaker, hearer, Informif(hearer, speaker, prop))
Informif(hearer, speaker, prop)
decomp2: Request(speaker, hearer, Informref(hearer, speaker, var, prop))
Informref(hearer, speaker, var, prop)
decomp3: Request(speaker, hearer, Informplan(hearer, speaker, goal, plan))
Informplan(hearer, speaker, goal, plan)

Fig 7: Speech Acts Revised

suggest a plan, declaratively or imperatively. The effect of a suggestion is that the hearer is aware of a (new) option. (This may be construed as a question of attention, if only attention is understood.) The Ask plan now includes a request for a plan suggestion, mediated by Inform-Plan which is a specialization of Inform. We will try to explain the gas station example in terms of a request for a plan suggestion. The time parameter is omitted for convenience.

3. A System for Computing Plan-Based Implicatures

3.1. The Architecture

The heart of this proposal consists in a set of constraints specifying implicatures that arise in discussion of plans like the gas station example, and a system in which to test them. The constraints correspond to the beliefs an agent needs to have to have a plan. In gross, if someone is cooperative they will tell you if they believe something is wrong with your plan. So if they help you with it, you can believe that as far as they know, what you intend is reasonable and effective. The specific conclusions below are exemplified using the Buy-Gas plan instantiated for the gas station example.

- Plan variables can be bound reasonably (time, e.g., agent can get gas in 10 min.)
- Plan constraints hold (e.g. the seller currently owns the gas.)
- Plan preconditions are likely achievable (e.g. agent has money or can get it.)
- Plan decomposition is effective (e.g. plan would fail without the goto step.)
- Plan effects will hold after plan is executed. (Agent will own gas.)

Problematic words like "reasonable" and "likely" will be defined away in a proper formal treatment. Also, the last constraint is more subtle than it appears. It is simple to infer in a limited domain like the Blocks World that a plan, when executed under its constraints and preconditions, will have the specified effects. In richer domains, the *Qualification Problem* arises: it is no longer possible to list every extraordinary circumstance that would cause the intended plan to fail. Since I regard constraints and preconditions as those circumstances that *typically* must be attended to in plan execution, it follows that unusual circumstances, whether finitely representable or not, are not represented in the plan structure. So in any implicature that the goals of a plan would actually be achieved by that plan lurks the assertion that the speaker knows of no happenstance that will cause plan failure[6]. Grice's principle of cooperativity allows us to shift the burden of the Qualification Problem onto the speaker. Thus we are not obliged to explain how the speaker isolates possible qualifications of a plan, but we are entitled to conclude that the speaker is unaware of any if none are mentioned. There is a certain asymmetry here which is hard to represent, namely, that there is some information best known to the person with the goal (as, whether there is money in his pocket). For now, we will pretend that this is captured by the distinction between constraints (external conditions) and preconditions

[6] In Pelavin's notation, ~BEL(speaker, ~EXECUTABLE(plan))

(controllable ones.)

The constraints all apply in cases where a plan was suggested or elaborated by a response (say, an Informref). Some common negative responses of the same flavor (e.g., denial of a plan constraint) will be handled by our system in the process of performing plan recognition on the utterance. Research on correcting misconceptions also yields sets of classes of non-positive responses that are based on plan structure. This literature attempts to provide computational methods for identifying false assumptions on the part of an expert system's user, and generating an appropriate response. In general these responses are very direct and easy to interpret: their propositional content embodies how the user must adjust domain knowledge or plans. In schemes like that of Quilici et al. [Quilici 87], McCoy [McCoy 87] or Pollack [Pollack 86] these propositions are expressed in terms of plan structure. Carberry [Carberry 87] and McCoy each also consider the expression of corrections in terms of the structure of object representations and the type hierarchy. Engineering the understanding of such responses would be a good exercise but seems to contain few research issues. The real problems seem to be with much less explicit responses and with failures of the cooperativity assumption, neither of which have been addressed to date. It seems reasonable that intonation might provide clues to non-cooperativity in some of its forms[7].

For the remainder of this proposal we shall consider primarily positive responses.

The plan-based approach has another whole set of constraints to offer, besides those above. Consider maxims Quantity 2 and Manner 3, which basically refer to efficiency considerations (don't use an inefficient method when an efficient one will do). These may be regarded as specializations of a general efficiency principle, similar to that used by Wilensky [Wilensky 83] for story understanding. Wilensky lists a number of such principles, including ones involving multiple goals and agents, and it would be worthwhile to investigate their relationship to the Gricean discourse principles. I propose this for discussion rather than implementation.

A system to demonstrate the operation of the first group of constraints is overviewed in Fig. 8. Its modules will be described below, but their precise specifications and interfaces remain thesis work. The system models an agent executing a plan, who must occasionally ask for help (information) in order to continue. The system starts up with (an input) plan or goal, and begins plan execution. When it needs further information, it begins a discourse plan. The system asks its question in the form of a surface speech act specification, and receives an input answer in kind. (It may be possible to interface a parser here.) Then the system uses plan recognition to processes the answer, including implicature calculation. It then continues discourse or domain plan execution, or gives up as appropriate. It is thus neither a planning system nor a question-answering system but a plan executor and a questioner. In addition to yielding a trace of its operations, the system will handle direct database queries.

Much of the static data that the systems needs is organized as an ISA hierarchy. ISA(X, Y) is the usual abstraction relation (without exceptions, for now). Domain and discourse plan schemas are

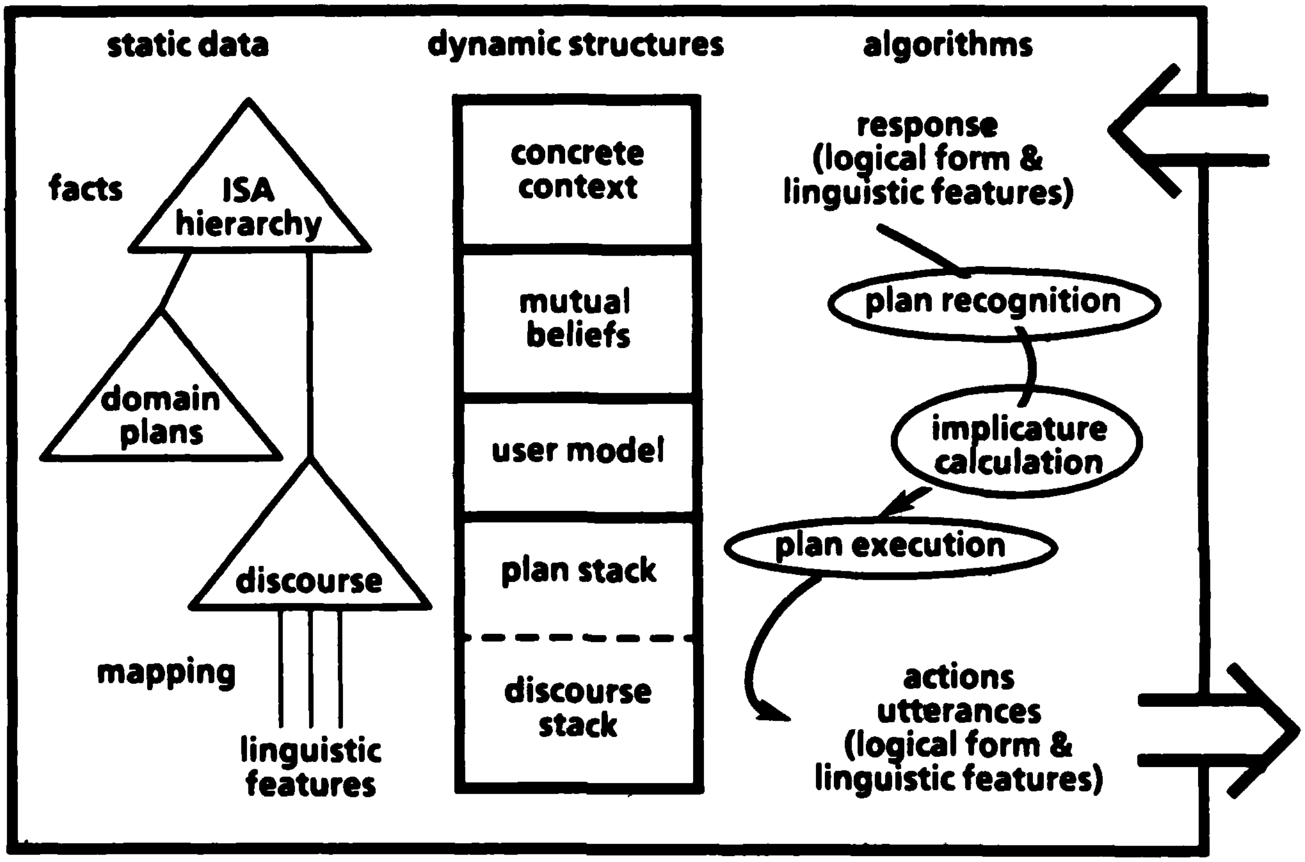


Fig 8: An Implicature Computer

part of the ISA hierarchy, as are object schemas. For plans in particular it will be taken to mean that any schema information for plans of type Y will also be true for plans of type X. Some general suggestions for organizing ISA hierarchies can be gleaned from Rosch et al. [Rosch 76] which presents a number of psychological studies of human categorization of objects. Concrete facts and other more general facts are represented as axioms in FOPC. There must also be provision for the grammar if parsing is to be done, and any other linguistic information. We had speculated at one time about a verb hierarchy, for instance. This part of the system depends on further research about surface utterances.

Dynamic structures important to conversational implicature include the plan execution stack, which is also used for discourse plans, a model of the hearer, a model of mutually believed information, and what I call concrete context. The plan stack is the usual basis for plan execution, and during discourse contains some of the information in Grosz and Sidner's intentional structure [Grosz 86]. It holds any active plans and indicates what action comes next. The hearer model holds beliefs and intentions that are attributed to the other agent, as they are identified. Some knowledge can be earmarked as mutually believed. The hearer model and mutual beliefs are a

[7]But I won't actually do any work on this for my dissertation.

potential bottomless pit: belief is an unsolved research problem and interesting failures of implicature can often be explained in terms of discrepancies in beliefs. Thus there would be a lot to be gained for implicature by implementing a sophisticated belief model. But I will build a simple one using available facilities rather than trying to build them on the fly as suggested by Wilks [Wilks 86] or conducting any extensive research on the subject. It may also be that the "concrete context" module will be subsumed by a good treatment of mutual belief; the point of putting it into the diagram is to remember that some non-linguistic context can be held out as particularly relevant to the conversation. Here too abide monsters.

There are several algorithms to the system, which invoke each other. Plan execution is the main algorithm, and consists in stepping through a plan in the obvious way. When it needs information in order to continue, it makes use of a discourse plan. The response is then interpreted using plan recognition (parsing first, if it seems feasible). Plan recognition makes use of abductive equality assertion and possibly weak matching, as discussed below. The overall process of understanding an utterance is as follows:

(1) Do plan recognition, on behalf of the hearer, of the system's own utterance (question). The result is a set of beliefs which are added to the hearer model. The plan recognition algorithm will be an elaboration of Kautz's work with abstraction. It is described in more detail below.

(2) Try to match the response against system's planned response. Note that this match, as all others to come, is abductive: if it is consistent to equate constants in order to make the response unify with the expected one, we will do so. The predicate DIFFER will be provided to allow explicit assertion of inequality. We will describe an abductive equality algorithm in more detail below.

(3) If that fails, try to match against parts of the mutually believed discourse plan. This means matching the response and its negation against each constraint, precondition, action, and goal in the discourse plan on the top of the mutually believed stack, and any discourse plans underneath it if necessary.

(4) If that fails, continue down the stack through the mutually believed domain plans.

(5) If that fails, repeat 2-4 with "relaxed matching." The simplest version of relaxed matching is to weaken the type restrictions on propositional variables by choosing their superiors in the type hierarchy. Arbitrarily more complicated schemes can be devised, depending on the exact structure of object representations. McCoy [McCoy 87] describes one such scheme, based on weighting the importance of each attribute an object can have. See also [Goodman 86]. This problem is worth some thought.

(6) If that fails, repeat using forward chaining or other such search techniques. One could start with the response and the predicates it contains, but the predicates in the plan stack are other possibilities. Perhaps something clever could be done here.

(7) Try plan recognition from scratch if that fails, again after Kautz.

(8) Give up if everything has failed. If something succeeded, compute implicatures according to the constraints described earlier and continue plan execution.

This might seem like quite a complicated context for testing the implicature algorithm, and in fact it is. The main reason for the complication is that there are any number of responses that can be made to a question, all of which license the implicatures described above. In order to handle a reasonable range of responses, we must be able to relate any of them to the system's plans. Another reason, of course, is that any process that people use to interpret responses is highly parallel, and the understanding that we have of parallel processing in general is much too weak to be incorporated here now without greatly obfuscating the theory. This is an interesting area for future (as in post-Ph.D) exploration.

The plan recognition algorithm developed by Henry Kautz is designed to yield as many *certain* inferences as possible about a plan, even when it can't be identified exactly. It relies on an abstraction hierarchy of plans, their decomposition relations, and two other predicates. END-EVENT(A) indicates that an action A is done for its own sake rather than in the decomposition of any other plan. BASIC-ACTION(A) asserts that A is an action that has no decomposition. It is primitive. On the assumption that all plans are known, plan recognition takes as input some series of observed basic actions, finds the plans that they each can be part of, abstracts upward to END-EVENTs, and selects the smallest subset of END-EVENTs that accounts for all the actions. Then any conclusions based on the END-EVENTs (and any more detailed plans that are uniquely identified) can be drawn.

A word is in order about abductive equality assertion, too. We developed a simple, efficient, but incomplete abductive equality checker for the Allen story understander [Allen 87]. This algorithm recursively checked type compatibility of structured objects' role values, asserting the corresponding parts to be equal if each pair of types was consistent (same type or one a subtype of the other.) Rather than checking every proposition about all these parts for consistency, only the explicit DIFFER was checked. Charniak [Charniak 86] describes a complete abductive equality algorithm which checks every proposition about the object and its roles recursively, and uses a depth bound to prevent infinite looping. Both algorithms check for provable equality first. A revised version will do a very shallow consistency check of propositions in addition to DIFFER.

GOAL: HAS(System, ?G*Gas)
METHOD: ?P*Plan

Fig. 9a. Domain goal

GOAL: HAS(System, ?G*Gas)
METHOD: ?P*Plan

GOAL: KNOWREF(System, ?P*Plan, GOAL(?P*Plan, HAS(System, ?G*Gas))
METHOD: Ask (System, User, ?P*Plan, GOAL(?P*Plan, HAS(System, ?G*Gas))

Request(System, User, Informplan(User,
System, ?P*Plan, HAS(System, ?G*Gas)))

Surface-Inform(System, User, ~HAS(System, ?G*Gas))

Informplan(User, System, ?P*Plan, HAS(System, ?G*Gas))

Suggest(User, System, ?P*Plan)

9b. Domain goal and discourse plan

3.2. Examples

Let's look at the gas station example in detail:

You are walking down a road when you see a car on the shoulder and a person approaching you with a gas can in hand, who says,

(1) a: I'm out of gas.

You reply,

b: There's a gas station around the corner.

The person is now entitled to conclude that as far as you know, the gas station is open, will actually sell the gas, and so on.

Utterance 1a is best regarded as a general statement inviting a suggestion[8]. The system starts with a goal of having gas, and no method of achieving it. This state is shown in Fig. 9a. Items such as ?S*Human are existentially quantified variables of the type following the asterisk. The others are constants: System, User, Station1 (a gas station), Location2 (a place), and Now+10min (a time in the near future). The system then selects a discourse plan, an Ask plan, in order to find out how to satisfy the goal. It requests an InformPlan (Fig. 9b). The Request is realized by a surface speech act of the form Surface-Inform(system, user, proposition) [-subjun, -if, -please], where

[8]A request for help, actually, but we have not yet developed a general schema for help that encompasses both suggestions and other forms of help.

proposition is the undesirable state of being out of gas. The system then does plan recognition to simulate the user's understanding. There are five different Surface-Informs that could be recognized, but this one doesn't match a known goal or an action but a state. Request **decomp11**. So decomposition links allow the system's simulation to correctly identify the Ask, and the goal is found in the proposition. Both the user model and the system's expectations are updated to expect the Suggest that implements the Informplan (the hearer model is not shown).

The user's response (Fig. 9c) is Surface-Inform(user, system, proposition) [-subjun, -if, -please], where *proposition* is that there exists a gas station at a certain location. This matches **decomp3** of the Suggest plan schema. Since the Suggest act was anticipated, and matched successfully, the other uses of a declarative statement are not investigated. However, the suggested plan must first be identified.

Standard plan recognition takes as input some action or series of actions, and performs some computation on the plan hierarchy to identify the plans that might be in progress. In processing this InformPlan to identify the domain plan, the system has at its disposal not actions, but a goal and the variable value given in the Informref. An extended Kautz-style plan recognizer would use the goal and value to identify the set of plan schemas that are potential matches. Suppose for a moment that only the Buy-Gas plan was found. It would be inferred first in the user model.

GOAL: HAS(System, ?G*Gas)
METHOD: Get-Gas(System, Station1, Now + 10min, ?G*Gas)

GOAL: KNOWREF(System, G1, GOAL(G1, HAS(System, ?G*Gas))
METHOD: Ask (System, User, G1, GOAL(G1, HAS(System, ?G*Gas))

Request(System, User, Informplan(User,
System, G1, HAS(System, ?G*Gas)))

Surface-Inform(System, User, ~HAS(System, ?G*Gas))

Informplan(User, System, G1, HAS(System, ?G*Gas))

Suggest(User, System, G1)

Informref(User, System, Station1, AT(Station1, Location2))

Surface-Inform(User, System, AT(Station1, Location2))

Fig. 9c. Plan system recognizes after user's Surface-Inform

moved to the system's plans, and then used to compute the following implicatures.

- There are a seller, a location, a time and some gas, each with reasonable value. (The variables in the plan have reasonable bindings.)
- The agent has some money or can plan how to get some. (The plan preconditions hold or can be achieved by plans.)
- The seller owns the gas, and both are at the gas station location at the time. (Plan constraints will hold at the time of plan execution.)
- The agent need only go there, hand over the money, and receive the gas. (Plan decomoposition is appropriate and workable.)
- Then the seller will own the money and the agent, the gas. (The effects of the plan will hold after its execution.)
- There isn't anything likely to interfere with this plan. (The effects of the plan will hold after its execution.)

In fact, though, there are two domain plans that meet the restrictions in the InformPlan, Buy-Gas and Steal-Gas. They are collapsed by abstraction into the Get-Gas plan, which is attributed to the hearer, and then the system in Fig. 9c. The Get-Gas implicatures are weaker, as follows.

- Shortly is ok.
- Agent can get there shortly.
- Agent can take gas.
- Gas will be there.
- Agent will have gas in the end.

The disjunction of the more specific information about the two specializations could be drawn too. Although Grice said that implicatures may be disjunctions, we would prefer to conclude only what is unambiguous and leave the rest as merely a basis for possible subsequent disambiguation rather than a full-fledged conclusion.

Let's consider what a system like this would do with some of our other plan-based implicature examples. In the library story, for example, the system models the student. The student has a plan to buy candy, which occasions a plan to get change, which leads to his Request. The response is an Inform carrying the *scalar* implicature[9] that he could have change, for alternate values of the bill. The plan-based implicatures are that his change and candy plans will work, for the alternate values. However, the librarian has licensed the implicatures for a different plan entirely, having assumed that the student wanted to make copies. When the student returns, the error is discovered. The system thus provides a framework in which problems of plan-dependent implicatures can be discussed.

---

[9]See Hirschberg for the computational account of scalar implicature. The system does not actually perform this computation.

3.3. Other Interesting Responses

For the gas station example alone, many interesting responses can be found that challenge the proposed system.

(15) There's an Amoco around the corner.

is handled in Step 2 just like the original response, provided that the system knows an Amoco has type gas station. If not, the system should be able to infer this via abductive rather than direct unification in Step 2. The behavior of the stacks and so on would be exactly as described above.

(16) There's a garage round the corner.

would fail the abductive match since garage is presumably known to be a sibling of the gas station type. It would not match against any part of the discourse plans (Step 3) or the domain plans (Step 4), but it would match against the planned response (in Step 5) when the type gas station is weakened to its supertype automotive. Thus the stacks can be popped as in Fig. 6d, and the system infers that while automotive places don't always have gas, the speaker believes that this one is likely to.

(17) Sorry, I don't know.

denies a constraint in the Ask plan. While it fails to match the planned response in Step 2, its negation matches a constraint not on the request but on the underlying Ask plan (during Step 3.) Negative matches fail to license implicatures according to our implicature constraints, although the stack is popped as before. An issue raised again by this example is that of clue words: when we hear "sorry" we already know that we should look for the denial of some constraint.

(18) Sorry, I'm new here.

is similar. But the matches fail all the way through Step 5; in Step 6 the required inference from being new to being likely not to know locations (constraint on that Ask) is performed. This axiom must be provided, of course. After popping the stack, the plan executor may want to attempt another Ask plan, but it should know that there's no point in asking this same speaker.

(19) It's a holiday.

is similar except that since the constraint on the domain plan is not achievable, the domain plan must be discarded with its goal unmet. A subsequent Ask plan might involve the same speaker, but a different query. Some examples that won't be handled are as follows:

(20) Stations are closed on holidays.

matches into the domain plan constraint in Step 4 using negation, but requires the added implicature (via relevance) that today is a holiday to be conclusive.

(21) Wouldn't you like to know?

This indicates a failure of cooperation, but without searching a list of templates for noncooperative responses I'm not sure how to recognize it[10].

(22) Hop in!

The speaker must be driving for this one to make sense. It matches a step in a decomposition of a specialization of a goto, and so requires some extra processing.

(23) You got a dime?

Totally unrelated plan. System actually gets this one if it has a plan to match it into.

(24) There's a Burger King round the corner.

Assuming that the system doesn't know about metaphor, it could either treat his like the Amoco example or be extended somehow to see that there has been a plan misrecognition.

4. The Proposal

4.1. The Research Plan

A complete dissertation would consist of

(1) A good specification of the system.
(2) The system itself.
(3) A set of example runs and a characterization of the general class.
(4) The set of plan implicature constraints.
(5) The response-matching algorithm.
(6) The abductive matcher.
(7) Extensive detail on illocutionary and surface speech acts.
(8) A review of the Gricean framework and maxims, formally.
(9) A demonstration that implicature requires plans.

Also the following are potential avenues for work, should time permit:

(10) Broadening coverage by adding e.g. temporal and location reasoning.
(11) How much can we constrain the parsing process via our surface speech acts?
(12) A good "weak" matcher.
(13) Extended discourse.

[10] Actually this particular noncooperative response is a query on the hearer's sincerity condition with interesting suprasegmental features, but though we could tackle it this way I'm not sure it would generalize.

(14) Focus and centering.

(15) More belief modelling.

(16) Use of concrete facts to direct plan recognition.

(17) More on accommodation.

(18) New techniques to direct inference for examples like those immediately above.

(19) Non-monotonic logical underpinning.

Each of the first nine points has been addressed in this document, and some parts have been implemented. A test system was implemented in Common LISP using HORNE [Allen 84] on the LISP machines in the summer of 1986, which performs the plan execution and implicature calculations. A version of the abductive matcher was implemented in the fall of 1986, as part of Allen's plan-based story understanding algorithm [Allen 87]. Further work will use HORNE's successor Rhetorical [Miller 87a, Miller 87b]. The set of examples, the response-matching algorithm, the Gricean framework, illocutionary and surface speech acts, and the importance of plans to implicature calculation are discussed in detail here. The main point is to get a system design that handles examples 15-18. Getting anything after #9 is a luxury.

In the first six months I expect to implement the domain plans[11], illocutionary speech acts, plan execution and recognition, abductive matching, and the implicature constraints. Note that a preliminary version of all this already exists. Any thinking about cooperative plans will need to be done before the implementation begins. At the same time I should finish off the model of surface speech acts as collections of linguistic features. And I may additionally consider one aspect of the problem: reference, accommodation, user modelling, or the role of concrete facts in implicature calculation.

In the second six months, it should be possible to implement surface speech acts (and maybe the other problem aspect). After that it remains only to write the dissertation. All implementation is in Common Lisp using the Horne reasoning system, and the Lisp Machine environment should facilitate work.

4.2. Related Work

The framework of the implicature computer is in many ways similar to that which Litman used [Litman 85] to handle fragments, ellipsis, and interruptions using "metaplans" that operate on plan structures. My debt to Diane is great, particularly since in this framework I too will get fragments and ellipsis for free. My implicature constraints and response matching process could be regarded as a family of additional metaplans, that allow an agent to restructure its plans and draw conclusions about them based on responses to its questions. Her system answers questions where mine processes those answers, relates discourse segments where I relate propositions, and

[11]Domain plans will be chosen from two domains to show the generality of the approach. The first domain will concern everyday actions like the gas station plans. The second domain will be naval search and rescue activities.

concerns itself much less with the searches involved in plan recognition and computing the secondary implications of utterances.

Grosz and Sidner [Grosz 86] have recently published their view of structures necessary for modelling discourse. It includes an intentional structure, which is a stack containing relationships among goals and actions, and an attentional structure, which is a stack containing names of objects that can be referred to. Our plan stack serves both of these roles, though perhaps less adequately. We have also concentrated on drawing conclusions rather than on perfecting the identification of surface references, trying more to exploit the structure of knowledge than the structure of discourse.

Carberry [Carberry 87] has used plan recognition techniques for the purposes of user modelling and misconception correction. The main goal of her plan recognition algorithm is to build up, over a series of utterances, a model of the user's plan. It is thus based on the relationships that actions have to one another in the plan structure. It does not make use of abstraction. Having built up a plan structure, it can use the plans to make sense of pragmatically ill-formed utterances, for instance, substituting a relevant attribute of an object for an inapplicable queried attribute. Again, my work concentrates more on unpacking the secondary conclusions that can be drawn from an utterance, and on finding the relationship of utterances to the plan, however indirect it might be. The misconception problem is an inverse to mine in that the misconception-correcting system has to make the user's queries fit its facts, while the implicature computer must find a way from its plans to the response it was given.

McCafferty, as we mentioned earlier, has proposed to import AI plan structures into the philosophical-linguistic literature of implicature. In fact the implicatures that he proposes to get from plans are similar to the constraints I propose, though he does not consider the Qualification Problem or Wilensky-style constraints. Neither is his theory embedded in any computational context, nor does it have anything to say about the links between plan structure and surface utterances.

There is a certain amount of work in story understanding that can be construed as an attempt to compute "conversational" implicatures [Charniak 85, Schank 77, Wilensky 83]. The general problem of story understanding differs from discourse analysis in two ways. First, the reader has no opportunity to ask for clarification, so the story itself must be sufficiently well expressed to communicate the writer's intent. Second, the reader effectively observes the characters through this carefully constructed keyhole, so there are no questions of, say, mutual belief. These differences aside, both problems involve analysis of intention and abductive reasoning. PAM, for instance, reads stories and produces natural language summaries of them from the characters' points of view, including motivations and inferred actions. In general, however, our inference rules are both more detailed in terms of plan structure and more general in terms of domain. We also include a theory of speech acts and their relations to agents' beliefs and surface utterances.

4.3. Conclusion

The problem of conversational implicature, which consists in trying to draw conclusions from utterances making use of context and world knowledge, could profit greatly from the plan-based approach to discourse. We have proposed here to try that approach within the framework of computational linguistics, by testing our postulated constraints and algorithms in a program to "understand" the answers to questions. We have also shown that there are a number of aspects to the problem that could be explored profitably.

References

[Allen 83] Allen, J., "Recognizing Intentions From Natural Language Utterances," in *Computational Models of Discourse*, Brady, M. and Berwick, B. (ed.), MIT Press, Cambridge, MA, 1983, 107-166.

[Allen 84] Allen, J. F., Guiliano, M. and Frisch, A. M., "The HORNE Reasoning System," 126 revised, Department of Computer Science, University of Rochester, September 1984.

[Allen 87] Allen, J., *Natural Language Processing*, Benjamin Cummings Publishing Co., 1987.

[Appelt 85] Appelt, D., *Planning English Sentences*, Cambridge University Press, 1985.

[Austin 62] Austin, J. L., *How to Do Things with Words*, Harvard University Press, Cambridge, MA, 1962.

[Carberry 87] Carberry, S., "Plan Recognition in User Modelling," *Computational Linguistics*, 1987.

[Charniak 85] Charniak, E. and Mcdermott, D., *Introduction to Artificial Intelligence*, Addison-Wesley, Reading, MA, 1985.

[Charniak 86] Charniak, E., *Motivation Analysis, Abductive Unification, and Non-monotonic Equality*, unpublished paper, 1986.

[Fikes 71] Fikes, R. E. and Nilsson, N. J., "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," *Artificial Intelligence 2*:3/4, 1971, 189-208.

[Gazdar 79] Gazdar, G., *Pragmatics: Implicature, Presupposition and Logical Form*, Academic Press, New York, 1979.

[Gibbs, Jr. 84] Gibbs, Jr., R., "Literal Meaning and Psychological Theory," *Cognitive Science 8*, 1984, 275-304.

[Goodman 86] Goodman, B. A., "Reference Identification and Reference Identification Failures," *Computational Linguistics 12*:4, 1986.

[Grice 75] Grice, H. P., "Logic and Conversation," in *Syntax and Semantics: Speech Acts*, Cole, P. and Morgan, J. L. (ed.), Academic Press, New York, 1975, 41-58.

[Grosz 86] Grosz, B. J. and Sidner, C. L., "Attention, Intentions, and the Structure of Discourse," *Computational Linguistics 12*:3, July-September 1986, 175-204.

[Grosz 87] Grosz, B. J. and Sidner, C. L., *Plans for Discourse*, Symposium on Intentions and Plans in Communication and Discourse, 1987.

[Harnish 76] Harnish, R. M., "Logical Form and Implicature," in *An Integrated Theory of Linguistic Ability*, Bever, T., Katz, J. and Langendoen, D. T. (ed.), Thomas Y. Crowell Company, Inc., New York, 1976, 313-392.

[Hirschberg 85] Hirschberg, J., "A Theory of Scalar of Implicature," MS-CIS-85-56, PhD Thesis, Department of Computer and Information Science, University of Pennsylvania, December 1985.

[Horn 72] Horn, L. R., *On the Semantic Properties of Logical Operators in English*, PhD Thesis, Yale University, 1972.

[Horn 84] Horn, L. R., "Toward A New Taxonomy for Pragmatic Inference: Q-Based and R-BASED Implicature," in *Meaning, Form, and Use in Context: Linguistic Applications*, Schiffrin, D. (ed.), Georgetown Univeristy Press, Washington, DC, 1984, 11-42.

[Karttunen 79] Karttunen, L. and Peters, S., "Conventional Implicature," in *Syntax and Semantics, v11: Presupposition*, Oh, C. and Dinneen, D. A. (ed.), Academic Press, New York, NY, 1979, 1-56.

[Lewis 79] Lewis, D., "Scorekeeping in a Language Game," *Journal of Philosophical Logic 8*, 1979, 339-359, D. Reidel Publishing Co..

[Litman 85] Litman, D. J., "Plan Recognition and Discourse Analysis: An Integrated Approach for Understanding Dialogues," TR 170, Department of Computer Science, University of Rochester, Rochester, NY, 1985.

[McCafferty 86] McCafferty, A. S., Explaining Implicatures, 23 October 1986.

[McCoy 87] McCoy, K. F., "Reasoning on a Dynamically Highlighted User Model to Respond to Misconceptions," *Computational Linguistics*, 1987.

[Miller 87a] Miller, B. and Allen, J., *The Rhetorical Knowledge Representation System: A User's Manual*, forthcoming technical report, Department of Computer Science, University of Rochester, 1987.

[Miller 87b] Miller, B., *Rhet Programmer's Guide*, forthcoming technical report, Department of Computer Science, University of Rochester, 1987.

[Nilsson 80] Nilsson, N. J., *Principles of Artificial Intelligence*, Tioga Publishing Company, Palo Alto, CA, 1980.

[Pelavin 86] Pelavin, R. and Allen, J. F., "A Formal Logic of Plans in Temporally Rich Domains," *Proceedings of the IEEE 74*:10, October 1986, 1364-1382.

[Perrault 87] Perrault, C. R., "An Application of Default Logic to Speech Act Theory," in *SDF Benchmark Series: Plans and Intentions in Communication and Discourse*, Cohen, P. R., Morgan,

J. and Pollack. M. (ed.). MIT Press. Cambridge. to appear 1987.

[Pierrehumbert 87] Pierrehumbert. J. and Hirschberg. J., *The Meaning of Intonational Contours in the Interpretation of Discourse*. Symposium on Intentions and Plans in Communication and Discourse. 1987.

[Pollack 86] Pollack, M. E., "Inferring Domain Plans in Question-Answering," MS-CIS-86-40. PhD Thesis. Department of Computer and Information Science. University of Pennsylvania. May 1986.

[Quilici 87] Quilici. A., Dyer. M. and Flowers. M., "Detecting and Responding to Plan-Oriented Misconceptions." *Computational Linguistics*. 1987.

[Rosch 76] Rosch. E., Mervis. C., Gray. W., Johnson. D. and Boyes-Braem. P., "Basic Objects in Natural Categories," *Cognitive Psychology 8*. 1976. 382-439.

[Sacerdoti 74] Sacerdoti, E. D., "Planning in a Heirarchy of Abstraction Spaces," *Artificial Intelligence 5*, 1974, 115-135.

[Sacerdoti 80] Sacerdoti. E. D., "Plan Generation and Execution for Robotics," Technical Note 209, SRI International. Menlo Park. CA. April 1980.

[Schank 77] Schank. R. C. and Abelson. R., *Scripts, Plans, Goals, and Understanding*. Lawrence Erlbaum Associates. Hillsdale. NJ. 1977.

[Searle 75] Searle. J., "Indirect Speech Acts," in *Syntax and Semantics, v3: Speech Acts*. Cole and Morgan (ed.). Academic Press. New York. NY. 1975.

[Sperber 86] Sperber. D. and Wilson. D. in *Relevance: Communication and Cognition*. Harvard University Press, Cambridge. MA, 1986.

[Thomason 84] Thomason. R., "Accommodation. Conversational Planning. and Implicature." *Theoretical Approaches to Natural Language Understanding*, 1984. (date?).

[Weischedel 79] Weischedel, R. M., "A New Semantic Computation while Parsing: Presupposition and Entailment," in *Syntax and Semantics, v11: Presupposition*. Oh, C. and Dinneen, D. A. (ed.), Academic Press, New York, NY, 1979, 155-182.

[Wilensky 83] Wilensky, R., in *Planning and Understanding*. Addison-Wesley Publishing Company, Reading, MA, 1983.

[Wilks 86] Wilks, Y., "Default Reasoning and Self-Knowledge," *Proc. IEEE 74*:10, October 1986.

END
DATE
FILMED
MARCH
1988
DTIC