In Paris, Swartout & Mann (eds) **Natural Language Generation in Artificial Intelligence and Computational Linguistics**, Kluwer Academic Publishers, Boston, 1991, pgs. 229-248. Originally presented at the 4th International Workshop on Natural Language Generation, held on Catalina Island, California, July 17-21, 1988.

ON THE PLACE OF WORDS IN THE GENERATION PROCESS

David D. McDonald[1]

July 1989

Abstract

This paper describes preliminary research on "lexical choice" in generation: the relationships between words and the representations employed by the speaker for reasoning and modeling the situation. We hold that the bulk of the variance that we see at the surface level in language has its origins very deep in the conceptual system. Consequently, most of the burden of lexical choice must be taken on by this internal representation, and not by the generator proper as is customary today.

This work is exploratory rather than comprehensive. It studies the design consequences of three examples of lexical choice, introducing the devices "lexical clusters" and "action chains" as part of the representational system that organizes the selection process. In the first example the choice follows directly from the speaker's categorial judgements. The next describes how the very same information can receive two substantially different lexical realizations depending on the speaker's attitude toward it. In the last a choice usually ascribed to the generator is reanalyzed as conceptual, leading to a simpler processing architecture. These studies lend support to the conclusion that the selection of key lexical items is the first step in generation, with the choice criteria taken almost exclusively from the conceptual model and intentional attitudes of the speaker.

[1] Author's address: 14 Brantwood Road, Arlington MA 02174-8004, (617)646-4125, Fax 648-1735
MCDONALD@BRANDEIS.EDU

1. Introduction

With only a few exceptions, generation researchers have paid little attention to the nature of words. Cumming's 1986 review of generation lexicons identifies many more open problems than it does accepted solutions. Marcus in Tinlap-III (1987) chides us for systematically trivializing the problem of lexical selection in the way we focus our research.

There have been good reasons for this inattention. A generator can do no better than the material it has to work with. This material is supplied by the application program that underlies and directs the generation process, i.e. the program that provides the motivations for the utterances the generator produces and grounds its processing in an actual situation addressing a specific interlocutor. As a consequence, the representation and model that the "underlying program" uses have an enormous impact on what the generator is able to do.

For the most part, the applications programs that have been used to date to drive generators have all employed the same representational style, a style where the conceptual model is built from a large number of very specific primitives which invariably have ready correspondences to individual English words or fixed phrases. For all intents and purposes these programs are already "thinking in words". There is very little for their generators to do in the way of lexical selection beyond reading out and artfully composing what has already been fixed within the underlying application program.[2]

Today, however, the linguistic competence of our generators is quickly exceeding the modeling and reasoning ability of our client underlying programs. We can begin to see rationales for complex texts but cannot find legitimate sources or motivations for these rationales in the programs we have to work with. If we are to make progress in generation we must find or develop programs with deeper conceptions of what they know and how they reason with it, conceptions that lead these programs to take on more complex perspectives and intentional states. This increase in complexity would remove application programs from the realm of "capital letter semantics", with its trivialization of lexical choice, and into designs where terms in the programs no longer map one to one onto the entries of an English lexicon but instead prompt substantial judgements about how they should be realized. A significant part of the research to develop these deeper conceptions will come down to exploring the nature of words.[3]

---

[2] On the other side of the coin are programs with representations based on a very small number of primitives, most notably Conceptual Dependency Theory ("CD"). Historically these have fostered a great deal of work on lexical selection in generation (Goldman 1974, Wilensky 1978, McGuire 1980, Hovy 1988). Since it is impossible to map directly to surface vocabulary when you are working with only, e.g., 13 primitive action terms, CD-based generation systems have always employed discrimination techniques capable of appreciating the context in which a term occurs and from that identifying the most precise word to use; the most thorough discussion of this style of lexical selection is in Goldman's 1974 thesis. Technically it is a simple matter to extend the discrimination process beyond simply looking at the semantic type of an action's arguments to considering rhetorical factors such as the speaker's point of view (Wilensky 1978) or affective goals (Hovy 1988). We will have nothing more to say about this kind of lexical choice procedure: Applications with this style of representation are increasingly in the minority (having been displaced by designs where the comparable generalizations are captured in class hierarchies or taxonomic lattices), and they are not part of any of the systems we are dealing with.

[3] We are ultimately not just concerned with the selection of individual words, but with a broad set of non-compositional information-bearing linguistic devices: collocations, poly-words, idioms, productive phrases, marked syntactic constructions, productive morphology, etc.---a set that may be collectively termed **linguistic resources** to emphasize their role as units the text planner can draw on in assembling an utterance. This is in keeping with the general presumption of generation researchers that "lexical" information includes far more than simply lists of words and their properties (see for example

This paper presents a few steps towards a model of the place of words with respect to underlying conceptual representations and of the process of lexical choice. This model is emerging from a body of ongoing research done in collaboration with Marie Meteer (BBN) and James Pustejovsky (Brandeis University). The paper will attempt to establish that lexical choice is primarily a matter of the model and attitudes held by the speaker at the time of speech, i.e. that the primary resolution of of options in decisions involving lexical choice occurs within the underlying program or directly reflects its categorizations and perspectives.

This in effect places words just outside of the generator proper---a possibly self-contradictory statement that requires some clarification. In generation research we talk about words in terms of "lexical selection" because we are focusing on the decision-making processes involved: why is one word chosen rather and another; what are the computational and contextual circumstances that define or constrain the available options; what is the character of the deliberations involved in making the choice; how is this choice represented once it is made. By saying that lexical choice directly reflects the categorizations and attitudes of the underlying program we are saying that no decisions of much consequence remain for the generator as the identity of the words is fixed by considerations that properly belong to the reasoning and modeling capacities of the underlying program rather than the linguistic capacities of the generator, e.g. inferential consequence, perceptual categorization, perspective, relative salience.

The generator does, of course, establish what words are available;[4] but since the mapping of information into words is based on criteria outside of the generator's scope, then its work with words will be largely confined to accommodating to the constraints the lexical choices carry with them.[5] Similarly, the generator is where the actual "act" of lexical selectino takes place since the use of words is not relevant to the underlying program's activities. The generator provides the impetus, but the decision criteria are primarily within the underlying application program.

2. The Relationship Between Words and Conceptual Structures: Feature matching or direct lookup?

A proper place to start in a discussion of lexical choice is with some examples of what people actually do. Of course since we do not have direct knowledge of the generation processes people use, some judgement is required when interpreting the texts we collect when doing an empirical study; however I think even shallow observations are very telling.

Hovy 1987 or Jacobs 1988) Differences emerge in how this range of resources is encoded and deployed, some making the "lexicon" primary to the point of subsuming the grammar, and others doing the reverse. As my present goal is only to broaden the readers conception of where and how lexical choice might be done and not to present a detailed architecture, I will have nothing to say about such design alternatives.

[4] Gaps in this competence pose important and methodologically useful problems for text planning in generation. These will not be considered here; but see Meteer 1988.

[5] Note, this discussion only applies to open-class, content words and information-bearing closed-class words ("above", "after"). Function words such as "to" or "the" we take to be introduced late in the generation process as the realization of grammatical relations and grammaticised semantic categories (such as definite/indefinite).

Consider the two excerpts below from a transcribed description by a person of the layout of her apartment. This data comes from a preliminary study done in collaboration with Allison Huettner and Penelope Sibun. A report on its background and methodology appears in Sibun, Huettner & McDonald 1988. The first excerpt is typical of these monologues overall: things and locations are described by simple NPs built on familiar common nouns; spatial relationships are given by clauses built on a regular pattern that the speaker fell into almost immediately when asked to do the task.

> *then, in the kitchen there's a large window which faces the backyard, with two smaller windows flanking it. And if we're facing towards the backyard now, on the righthand side is a sliding glass door, and a few feet from that is a smaller window, towards the living room.*

The second excerpt comes at a point when the speaker has had to name a place in the apartment that we suspect she has never needed to name before. It is not a conventional place-type in a house: neither quite a room nor a hallway but a mixture of both, and not a place that the people in the household ever did anything in---just a spot that they passed through on their way elsewhere. This speaker, like virtually every other speaker in our corpus for this house, hesitates, then composes a characterization of the place using very general vocabulary and specializing phrases.

> *Then in the righthand doorway, we have like a um we have a hall--large hallway that leads into the kitchen. ... and we walk down that wide hallway, which is almost a room in itself. There's a closet, on the lefthand wall, in the lefthand wall lets say of that hallway.*

The issue I wish to illustrate with these excerpts concerns the character of the work that is done each time a lexical choice is made: is it a calculation or a memory lookup? We start from the hopefully uncontroversial premise that naming is primarily a matter of characterization. The speaker called her kitchen a "kitchen" because it fell into that category of room-types by warrant of its appliances, furniture, primary use, etc.[6] Her hesitation over what to call the room/hallway in the second excerpt reflects her uncertainty as to what kind of thing it is. (Others used phrases like "the L-shaped room".)

How did the speaker decide what she was going to use as descriptions of the rooms? She surely did not have to deliberate over how to classify her kitchen; rather she simply recalled the long standing association between that specific room and the name "kitchen" that goes with that room-type. By the same token, when a category was not immediately available in the case of the odd room/hallway, the speaker was forced to think about what room-name best fit the space's properties before she could give a satisfactory description. (Notice her restarts and use of the hedge "like".) Once the categorization was established, however, she is at least in part drawing on her memory of what she said before, since she uses the subsequent reference forms "that" and the dropping of modifiers. (Notice though, that the modifier has changed from "large" to "wide". Whether this reflects a

[6] The utility of characterizing a room as, e.g., a kitchen is of course socially mediated. She calls the room a kitchen in no small part because her fellow interlocutors understand her when she does so. She might be less successful with, say, "breakfast nook" unless that was a socially effective characterization in her household. An additional consideration is the extension of basic descriptions to provide discriminating information. A large house might have an "upstairs kitchen" and a "downstairs kitchen" for example. The choice of modifiers in a case like this has less to do with categorization than with what attributes the speaker can expect to be salient to her audience. (For a discussion of the range of issues in noun phrase descriptions see Appelt 1985).

reclassification or the adoption of a collocation particular to hallways is difficult to determine.)

I take observations like these as evidence that one of the prevalent themes in research on lexical choice is misguided. This is the idea that lexical choice is a "feature matching process" (see for example Levelt & Schriffers 1987, Nirenburg & Nirenburg 1988). In such a process, the items that serve as the conceptual sources of the individuals and relations to be expressed are characterized for the generator as a vector of abstract features and values. (Whether this property vector is also used internally by the reasoning system is seldom clear. If it were, it would strengthen the case for this kind of design.) The generation lexicon is taken to consist of a set of lexical entries that are characterized in this same currency of features and values, and the lexical choice process is a matter of looking for a "best match"---finding the lexical entry whose property vector values are closest by some metric to those of the item to be expressed (Nirenburg & Nirenburg 1988).

The feature matching model of lexical choice has a number of potential technical problems that must be settled satisfactorily before it can be adopted: Is the entire lexicon being searched every time, or just a subset? On what basis do we determine which properties are relevant in a given case? Does the processing initiative come from the conceptual objects or the lexicon? What communications bandwidth and processor architecture is being assumed? Can one be sure that the algorithms will run in comparable time for all normal cases?

But even ignoring these technical problems, there is an a-priori matter that, to my mind, makes the entire enterprise very dubious. It is one thing to say that some sort of comparison of properties and prototypes takes place the very first time that we take notice of an object or have occasion to reify it or refer to it---such a process is needed to do categorial perception. It is another thing entirely to claim that after we have arrived at a categorization we proceed to forget what we have learned and to redo our calculations each and every time the object is used. More likely, we cache the calculations in memory, associating the mental object with the linguistic resources that we selected for it. On subsequent occasions, rather than go through the best match calculation again, we draw on this direct link that we have saved, and make our "choice" on the basis of table lookup rather than judgements from first principles.

In all of this, the crucial question is how we design and optimize our architecture for lexical selection. Should it be for best match calculations, which if I am correct are only needed the first time an object is considered, or should it instead be for the retrieval of long-established associations? I personally opt for the later alternative, and thus prefer direct mapping, "dictionary" based architectures (McDonald 1983, McKeown 1985), where the objects that are referenced by the text planner or the underlying program are already tied to a specific entry or class of entries in the lexicon, either as individuals or because of their conceptual class. In this kind of architecture, lexical choice is not a matter for the generator, since in normal speech the choice that goes on is at a conceptual, rather than a linguistic, level.

This is not to say that feature matching should be ignored. Like language learning, it is a capacity that will have to be included in any complete language system. Furthermore, it may well be used frequently in some domains, for instance diplomatic announcements or stock market reports (Kukich 1988). In domains like these there may be relatively few recurring individuals but a great deal of global summarization via a small set of conventional phrases. A phrase like "trading was moderate" reflects a context-sensitive judgement that summarizes a great deal of numerical information using a sublanguage that its intended audience is very sensitive to.

In summary, the point of this section has been to argue that the typical means by which an item's lexical realization is determined is just a simple memory lookup. Proponents of lexical selection strictly by feature matching must explain why prior decisions cannot be stored and recalled but must be recalculated each time---this will not be an easy task. Given that some kind of search through a feature space does, of course, occasionally occur (as the example of the previously unnamed room illustrated), then the next question is whether this is a standard enough operation to imagine that provisions have been made for it within the core generation process, or whether, as I expect, it will turn out that these new phrases are formed by an off-line calculation that draws on central, non-linguistic mental resources and is not part of the generation "linguistic module" in the sense of Fodor (1983).

3. "Lexical Clusters" to realize "sublexical" conceptual units

Nearly all generation research has been done for application/reasoning programs that were built by others and designed without regard for the needs of the generator. As a result, there are almost no proposals, let alone answers, as to how words and other linguistic phenomena might be deeply integrated with the internal representations that support reasoning.[7] That such an integration is possible, even preferable to facilitate reasoning, is an attractive possibility to many people, since they are drawn to the Sapir-Whorf hypothesis that the conceptual structures of one's language and one's thoughts are intimately linked. If the two are indeed linked, then generation becomes largely a problem of realization rather than translation, and a great many potentially insurmountable problems go away.

In this section, I will briefly lay out a design for lexical selection based on an underlying program that is being built from the ground up with the needs of generation firmly in mind. James Pustejovsky and I are building a reasoning system, called "Jeeves", that will act as an appointment secretary able to monitor one's everyday activities and look for conflicts. Our point is not to develop a competent time manager for its own sake (though that would certainly be nice), but to use the program as a laboratory for studying how deeply one can sensibly embed words within a computational model used for commonsense reasoning. We selected the domain of planning everyday activities because we will be making use of Pustejovsky's theory of Aspect Calculus (1988), and that domain has a heavy concentration of action verbs and temporal expressions.

---

[7] There is of course the trivial relationship between words and internal representation that is all too common in today's programs, namely identity: Applications programmers have a proclivity for using ordinary English words for the symbol names in their systems, and with a little artful programming these symbols can be substituted for variables in canned print statements to produce quite realistic output texts. The shallowness of the conceptual model behind these symbols, however, makes these systems quite brittle: If the utterance is made outside of the very specific context that the programmer had in mind when the print statement was written, it is almost inevitable that the words will not be used with their correct senses or connotations.

Pustejovsky and I are concentrating our study on nearly synonymous utterances such as these:

> *You can only stay until 4.*
>
> *You have to leave by 4.*

The "synonymy" of this pair of texts is due to the fact that they communicate largely the same information---each entails the other. At the same time they have a difference in perspective or connotation that one can begin to see by exploring specific scenarios where they could be used. As an example, imagine that you are spending the day at DEC consulting. You would usually leave when everyone else does at 5:30, but today you want to run an errand to pick up some film that you've left for developing at the Coop (a Cambridge department store). When you give this change in plans to Jeeves, it proceeds to work through the implications, letting you know if there are going to be any problems. In this particular case, given the distance to be covered, the closing time of the store, and the uncertainties of rush hour traffic, Jeeves might conclude that if you're going to get to the Coop before it closes that you're going to have to leave by 4:00. If (like just now) the attention is on the errand, it feels most natural to us that Jeeves express this using the *leave* phrasing; if, alternatively, attention were on the ongoing activity (say Jeeves already knows about the errand and you tell it you want to attend a lecture at 3:00), the *stay* phrasing feels better.[8]

3.1 "Sublexical" conceptual sources for words

In exploring these two alternative phrasings as a problem in lexical selection, we first considered what their source within the reasoning program (Jeeves) could reasonably be. Given our goal of experimenting with the Sapir-Whorf hypothesis, we wanted in particular to see how close to the surface wording the source concept(s) could be while still being an effective and general part of Jeeves' mental repertoire. In this case, as in almost all of the others we have analyzed, we were drawn to the early conclusion that the object best identified as the source was in fact **sublexical**. This is to say that it encodes less information than is required to specify a word or other atomic surface-level linguistic resource---to map it directly to a word would amount to committing to more specificity than the object warrants in isolation from its situation of use. Only when taken in conjunction with other information in the program's state is it possible to determine which real natural language word or other linguistic resource to use.

To make this clear we must first look in more detail at Jeeves' operation in this scenario. Jeeves' reasoning is based on a set of highly schematized specialists each able to reason about a different class of activity. The specialists communicate via a model of the user's schedule that serves as a shared blackboard, and are organized in a specialization hierarchy. The specialist for "picking up photographs at the Coop" is, for example, a specialization of "running an errand at a department store", which is in turn a composition of "errand" and "activity at a department store", which are themselves ultimately specializations of "activity".

When you inform Jeeves of your intention to pick up the photographs today, the agent for this errand is instantiated, its parameters are bound (e.g. prior location is DEC Marlboro,

---

[8] We believe that it is not especially important whether we are correct in the particular connotations that we ascribe, so long as there is consensus that such subtle differences do exist. In any event our approach to such analyses is always to ground them in concrete and plausible scenarios such as this one. As a result, we expect that these connotations, while possibly very bound to their contexts, should nevertheless be realistic.

default time is "end of the work day"), and the calculations are done to determine the implications of the errand for other, already scheduled, activities. When this is finished, the agent posts itself on the schedule blackboard. Part of the posting is an object---the source of our example utterance---that embodies the conclusion of the agent that is most important to the other scheduled activities, namely the time at which the errand activity has to start if it is to succeed. For convenience we can call this object "transition-at-4pm".

Transition-at-4pm contains information that identifies the time, 4 pm, as the value of an object of type "necessary moment of transition". The object has its parameter values set to suit this instance, e.g. the transition is between the two specific activities: the consulting day and the errand, and it is "necessary" because otherwise there won't be time to complete the errand. Being a constraint, it will activate a sentinel (Rosenberg 1978) associated with the pending consulting activity, which in turn will note that that doing the errand implies ending your day at DEC prematurely. Depending on where we set Jeeves' thresholds, this sentinel's activation could be enough to raise the significance of the object above the level needed for Jeeves to bring it to your attention. A inline call would then evoke the generator with the instruction to express the information packaged in the object.

Tacitly this amounts to an indirect speech act since it will have the impact of making you consider whether you really want to run the errand today or wait for another time. It would invite responses such as giving Jeeves an adjustment to your plans, asking *why?*, or starting an exploration of variations on the parts of plans that you knew had been assumed (e.g. "*but suppose I took route 2 instead of the turnpike?*"). Any proper treatment of the utterance as a speech act should include this kind of knowledge about its impact on subsequent discourse.[9]

To get the most utility out of transition-at-4pm, the object is designed to not just be the statement of a constraint between the two activities, but also the representation of their respective start and end times. (Under this design, rather than the start-time slot of the errand having the value 4 pm, a object of type clock-time, it has as its valuetransition-at-4pm, an object of type necessary-moment-of-transition. The object is effectively a variable---revised calculations may give it different values without changing its identity---and it has the value 4 pm.[10]) Since in this case the transition between them involves a physical change of location, the categorization of 4 pm as a start/end time can be specialized (subsumed) by a categorization in terms of stay and leave, which is what makes our target utterances relevant. It is this multiple categorization that is largely what is responsible for my speaking of the object as being sublexical: If an object can have multiple realizations depending on how it is classified (which in turn depends upon how it is being used at that moment), then none of those realizations can be justified outside of a classifying context.

3.2 The mechanics of selection from a lexical cluster

The act of making the lexical selections that realize transition-at-4pm takes place within the generator after Jeeves has passed it the object. The selection is essentially the first

---

9 At the moment, however, Pustejovsky and I do not anticipate doing any research on this sort of discourse control problem, and will just have a specific call to the generator as part of Jeeves' hardwired code. When we do take up this problem again it will likely be along the lines sketched in McDonald et al. 1986 and be done in collaboration with Philip Werner.

10 A data structure with such a multifaceted "personality" obviously does not have a simple implementation. This paper is not the place to go into any depth on the design. Suffice it to say that we are using a design where objects like our example are first class entities in the representation (meaning that they can be linked to other such objects by explicit functions/relations) and are composed out of other such objects which occupy named "slots" as in a conventional frame-based notation.

thing that the generator does, since we see the identity of the selected wording as the chief linguistic constraint on the rest of decisions that the generator will make. By making this most constraining decision very early, we are able to insure that the process overall can be done indelibly since the implications of the decision will be available to direct or modulate all later decisions.

I will focus here just on the lexical selection mechanism we are proposing. Our design for the early stages of the generation process overall---what is generally termed "text planning"---is otherwise in flux. What had been a simple matter under earlier, dictionary-based, designs is now very intricate. A short description of some of what we have been doing in text planning can be found in McDonald & Meteer 1988 and Meteer 1988 and 1989.

The object transition-at-4pm is a reification of a body of relationships among other objects in Jeeves' representation of the user's schedule and its own background knowledge. It was formed by successive layering of information (functional composition) in the course of the errand agent's calculations. Since it is a composite object, its realization by the generator will be compositional, and in this case will deliberately mimic the composition of the two target utterances. The two utterances, "*you have to leave by 4*" and "*you can only stay until 4*" are both statements of actions by the user (*you*), are both marked as necessary (*have to*, *can only*), and are both grounded on an action verb-preposition pair (*stay until*, *leave by*), which takes one argument (*4 pm*). We will look here just at the process of choosing the verb-preposition pair, since this is the part of the utterance that involves a lexical cluster.

Given this compositionality, the precise source for the verb-preposition element of the utterance is not the object transition-at-4pm as a whole but a sub-object in one of its fields. For convenience, we can refer to this object as StayTill/LeaveBy. Following our design rule that the association between a object and its realization is a direct link rather than a matching process (recall Section Two), there will be a long term link from StayTill/LeaveBy to an entity we are tentatively calling a **lexical cluster**. StayTill/LeaveBy is a permanent part of Jeeves' knowledge base, used in many rules and rule schemas; the instance object specific to this particular session with the user is transition-at-4pm. Transition-at-4pm could not have a permanent link to a lexical cluster (or any other kind of generation structure) since it only came into existence during this session. Its realization is controlled by links to its class rather than to it as an individual.

The process of lexical choice goes at follows. The generator looks up and activates the lexical cluster for StayTill/LeaveBy: The cluster examines the context (see below), establishes what it is about this context that will dictate its choice, and makes its decision, returning a specific lexical element that the generator incorporates into the appropriate place in its growing linguistic plan for the utterance.

The lexical cluster that StayTill/LeaveBy is linked to is specific to the pair *stay until*, *leave by*, returning whichever of these two fits the current context. There would be a comparable cluster for every permanent object in Jeeves that denotes a sublexical body of information: *go*/*come*, *buy*/*sell*, *send to*/*receive from*, etc.

As a point of contrast, the kind of decision made by a lexical cluster is very different, as we see it, from that needed to choose between, say, "*you have to leave by 4 or you won't have time to do your errand*" and just "*you have to leave by 4*"; or to choose between "*you have to leave by 4*" and "*you have to leave in 10 minutes*" (spoken at 3:50) or "*you have to leave soon*". Presumably, these other classes of decisions are amenable to general, structural accounts like the one are making for these lexical clusters; we have yet to study them in any detail.

Functionally the decision procedure of a lexical cluster is a discrimination net that tests Jeeves' state. Our goal in designing the tests is of course to capture as much generality as possible, and to this end we are employing an analysis of *stay until* and *leave by* that is based on Pustejovsky's theory of aspect calculus (1988). Briefly this theory postulates an algebra of events and process types for temporal aspect akin to Vendler's. Unit events combine into X-bar like trees, where position in a tree carries general entailments and presuppositions that follow from the shape of the tree rather than the specific events it organizes.

In an aspect calculus analysis, the verb *stay* corresponds to a tree whose root is a node of type process whose daughters are some unspecified sequence of simple events. The phrase *stay until* corresponds to a larger tree built on top of the tree for *stay* that is rooted in an accomplishment and whose daughters are the tree for the stay process followed by an event that is anchored to the moment in time that is *stay until*'s argument. Similarly the analysis of *leave by* is a tree whose first level constituents are an event anchored to the moment in time when you must leave followed by a tree for a process. In our present example, these two time-anchored events correspond to the very same object, transition-at-4pm, and the flanking processes correspond to the consulting activity (for *stay*) and the errand activity (for *leave*).

To make its discriminating tests, the lexical cluster must first establish the just-mentioned correspondences, which it does by following out the links from StayTill/LeaveBy via transition-at-4pm to the two activities (recall that transition-at-4pm is simultaneously the end point of the consulting activity and the starting point of the errand). Making the choice now comes down to determining which of these two activities is more salient given the point in Jeeves' code from which the generator was called on this occasion. For instance if the motive for informing you of "transition-at-4pm" came from the errand agent announcing a new thing that you now had to do, the errand would be more salient (this comes down to it being the activity in control at the moment of speech) and the utterance would come out as *you have to leave by 4*. Alternatively if the motive had come from the sentinel posted by the consulting activity, perhaps because its rules say that upon reflection you might not want to leave earlier than usual, then the utterance would come out as *you can only stay until 4*.

To summarize this section, our approach has been to work backwards from an instance of a lexical perspective pair in a concrete scenario, in this case "stay until"-"leave by". We developed a model in our scheduling domain that relates the difference in perspective to differences in the state of the scheduling program. Lexical choice is done by following a permanent link from a "sublexical" internal object to a "lexical cluster", which is an organization of alternative lexical choices into a discrimination net that tests the state of the underlying program to determine what is most salient and therefore which lexical choice should be made.

4. Choices made in the underlying program rather than the generator

In a conventionally organized generator, lexical and syntactic choices are made independently by different modules that are not active simultaneously. In these generators, a quandary arises from the possibility of close interaction and dependency between the two kinds of linguistic resources: Choices involving the two kinds of resources may be difficult to serialize consistently or may sometimes lead to backtracking--a markedly more complex control structure than one would like. (See discussion in Danlos 1984). To my mind part of this quandary has been an artificial limitation of the discussion. Lexical and syntactic alternatives are taken to be adjudicated only within the generator; we seldom consider the possibility that the solution could lie in the choice being made in the underlying program before the generator runs.

An instance of this limitation occurred in a recent workshop.[11] It comes in an argument made by Sergei Nirenburg in favor of a highly distributed model of the generation process as an alternative to the conventional design. He used as examples the two phrases below, both expressing the idea that at some time in the past the speaker had an intention "to go" but it didn't come to pass.

" *I planned to go . . .*"
" *I would have gone . . .*"

Nirenburg regards this as a case where the same conceptual unit may be realized either by lexical means (*planned*) or by syntactic means (*would have*). He then draws on a principle that is common to most theories of generation (indeed, to most theories of deliberate action of any kind), namely that for two different outcomes to be treated as equal alternatives in a decision process, they must be formally accessible at the same time within a single choice point. (See the recent discussion of this issue in Bateman 1988.)

If we provisionally accept Nirenburg's claim that the two alternative realizations are indeed properly characterized as involving lexical and syntactic resources respectively, then this data, plus the principle of simultaneously available alternatives, leads to the architectural claim that lexical and syntactic decision making must intermingle freely in a common phase of processing. Such a claim would mandate markedly more freedom in interleaving lexical and grammatical decision making than some researchers (such as myself) would allow. Yet without this relatively free mixing of the two kinds of decisions, a realization component would not be able to present both realizations simultaneously as alternatives for the text planner to choose between. Drawing on the support of this argument, Nirenburg (1988, 1989) proposes a highly distributed model for language generation where such interleaving is facilitated.

4.1 Two objects, two realizations

While it is true that these two phrases are "synonymous" in that they do communicate the same information, they are of the same type semantically: one is a possession ("I had a plan to go"), the other a state ("I might have gone but I didn't"). This is problematic, since it is hard to imagine that any general (i.e. non-idiosyncratic) linguistic class of alternatives could be motivated that would include both a possession and a state on equal terms, regardless of the generator's architecture.

This difficulty and others suggest that we look for other ways of analyzing the two phrases. I believe that rather than give up the otherwise successful premise that lexical choices precede syntactic ones,[12] we should look closely at the presumption that the two

---

[11] The 1987 "Natural Language Planning Workshop" sponsored by the Northeast Artificial Intelligence Consortium at Minnowbrook Conference Center, Blue Mountain Lake, N.Y. Proceedings available from NAIC c/o the Department of Computer Science, Syracuse University.

[12] More precisely, the premise is that for the unit of information encompassed by an elementary surface structure tree in a Tree Adjoining Grammar (Joshi 1985) the first and most influential choice is the lexical realization of the phrase's head. This selection determines the family of trees that can be employed

phrases should be seen as having exactly the same source in an underlying program. If the case can be made that the actual difference between the two is not the choice of linguistic realizations but rather a matter best appreciated before realization is begun, then the argument evaporates: There would no longer one object with two realizations, but two objects each with its own independent realization. (Note that it does not particularly matter what may be the case with the objects in an application program that we could take off the shelf today. This paper started with the assertion that we should treat the modelling and representational techniques of today's programs as suspect---too shallow in their treatments to support the delicacy and precision that we see in how people use language. New techniques are needed, and the demands imposed by sophisticated generation systems will play a significant role in their development.)

Certainly there is no reason to take it for granted that a reasoning system will always represent ostensibly synonymous texts with the same internal expression. It is true that early parsing systems would collapse "unimportant" text variations to the same "canonical form" (e.g passive-with-by-phrase clauses collapsing to the same internal expression as active clauses). But the reasoning performed by these parsers' backend applications was anything but sophisticated. Today we appreciate that the nuances texts can convey cannot be reasoned with unless they are reflected in the reasoner's representation. If it means something slightly different to the reasoner "to have had a plan to do something but not been able to carry it out" than "to have not done something", then there must be two different expressions in the underlying program. No doubt the expressions would be closely related and might literally share common parts, but some difference in categorization or internal makeup must be present if the difference in information content is to be consequential.

4.2 "Canonical action chains" to carry shared inferences

Assuming for the sake of argument that the *planned to go* / *would have gone* texts do correspond to different internal expressions and that the choice between them is a matter of promotion/selection rather than realization, a new question arises. How does the speaker/reasoner know that the texts are synonymous, i.e. that they can lead to the same inferences when communicated to a suitable audience? It is important to know this in order to carry on a fluent discourse: The response to what one says may be couched in a different way that one's audience assumed was an available alternative, yet it must still be recognized as cohesive. For instance suppose that one said "I would have gone on vacation, but there was just too much work to do" and got back the reply "Oh, where had you been planning to go?". The reply must be recognized as referring to what had just been said, which means that the close relationship between the two different surface events must be represented somehow.

One way to do this is to posit cognitive structures incorporating both alternatives that are mutually known by speaker and audience as a matter of common sense and shared experience. These structures would relate the two internal objects corresponding to these

---

(what in Mumble-86 is known as a "realization class", see Meteer et al. 1987), and with it the possibilities for expressing arguments, temporal information, existential status, multiplicity, etc. Note that this premise does not, however, carry with it any presumptions about the relative timing of realization decisions involving sets of these units severally or collectively (e.g. within complex sentences or paragraphs), nor any about the timing of the realization of information units syntactically embedded within an elementary tree (e.g. NPs embedded in clauses or some kinds of modifiers inside NPs---for instance the fact that some subunit will be expressed as a pronoun might well be known prior to realization of the unit as a whole).

texts, as well as related ones, all as instances of a general structure. Upon hearing one of the texts the audience would recognize and bring to mind the structure it was part of and thereby have access to any of the information that they would have gotten had they heard an alternative instead. Since the speaker knows that this will happen (through presumed mutual belief), he is free to use any particular item within the structure that he chooses, and can thereby communicate the nuances that come from deliberately holding a specific perspective.

As a candidate for the cognitive structure that would be needed in this case, I propose that our mental models include what might be called **canonical action chains**, script-like structures that represent customary sequences of mental states and actions. In this case the chain might be

[1] "wanting <to go>" -->
[2] "planning how <to go>" -->
[3] "intending <to go>" -->
[4] "doing <to go>" -->
[5] "having done <to go>".

The chain would be parameterized by the specific action and actor involved, and should include provisions for action failures or unforeseen contingencies as overlays to the default chain of events. In this example we had indeed "planned to go" (stage 2), and if nothing had intervened "would have gone" (stage 5).

The idea is that given any step in the chain, an audience will automatically infer the whole chain. Armed with this inference rule, a speaker can choose most any step of the chain as the basis of the utterance and expect to have an equivalent effect on the hearer. We would then look to other aspects of the speaker's intentional state to find the motivation behind the choice in a given case; for example this could be the relative salience of the steps to the internal agent that prompted the speech, as in the previous section.

There is obviously much more to be developed before a notion like canonical action chains could be accepted as coherent or useful (let alone psychologically plausible): more examples must be worked out, the particulars formalized and implemented, and evidence accumulated that the behaviors we would expect such a mental structure to facilitate in fact occur. Consequently my point here is not to convince anyone that some or another conceptual representation belongs in every application program, but simply to illustrate that we have many more options open to us as analysists than we tend to consider. We should not automatically assume that the difficulties we meet in doing our generation research should be solved within the generation process: a great deal of the variance that we see at the surface level in language may have its origins very deep in the conceptual system.

5. Concluding Remarks

The title of this paper asks what is the place of words in the generation process. As a field we have put off asking this question for a long time, probably for excusable reasons, but if we are to make further progress, particularly in the area of text planning, then we cannot put it off any longer. My answer to the question, at least for the moment, is that while the words per se are presumably linguistic elements and as such part of the generator, the criteria governing their use are dictated by the underlying program (mind) that drives and motivates the generator. This applies, furthermore, not simply to words but to linguistic resources of all kinds: collocations, idioms, marked syntactic constructions, etc., to the extent that these are not grammatically conditioned---which is only to say that there are other kinds of atomic linguistic packagings for meaning besides words.

This paper has argued for this view by example, showing that in the cases considered, the simpler and most direct designs all involved dependence on information from the underlying program's model and intentional state, rather than the linguistic state of the generator. This is in keeping with the idea that lexical choice is the first thing that the generator does, i.e. before it has established any substantial linguistic state.

For overlearned, highly familiar items like kitchens and keyboards, we argued that there was really no "choice" at all, the decision being implicit in the categorization. For the case where there appeared to be a linguistic decision required between syntactic and lexical alternatives, we argued that a natural conceptual representation, the "action chain", permitted a more motivated decision, again at a conceptual rather than a linguistic level. In the most elaborate example, we argued that the internal object whose information was being expressed was intrinsically "sublexical", and so could not receive a realization without also incorporating information about the speaker's attitute toward the object.

All of this of course only scratches the surface of the problems presented by lexical choice: There has been no discussion of the impact of prose style, or of the issue of what level of categorization to use (e.g. "Mac-II" vs. "computer"), to mention just two things. But I hope that an overall message has come through: For too long we have tended to focus our research on linguistic issues and not on mental models, and so have neglected the possibility that the input our generators receive may be conceptually far more refined and richly structured than what they are getting presently. This means we have missed opportunities for more elegant and extensible solutions where the underlying program's attitudes and models carry as much weight or more as the actions within the generator, thereby permitting many decisions to be motivated by general principles taking into account the system as a whole, rather than simply stipulated within the generator alone. Unless we collectively spread our focus to take in these models, we will not arrive at any satisfactory computational theories of lexical choice.

6. References

Appelt, Doug (1985) "Some Pragmatic Issues in the Planning of Definite and Indefinite Noun Phrases", Proceedings of the 23rd Annual Meeting of the Association of Computational Linguistics, July 8-12, Univ. of Chicago, 198-203.

Cumming, Susanna (1986) "The Lexicon in Text Generation", Workshop on Automating the Lexicon, Pisa, Italy; available at a technical report from Information Sciences Institute, Marina Del Rey, California.

Danlos, Laurence (1984) "Conceptual and linguistic decisions in generation", Proceedings of COLING-84, Stanford University, July 2-6, 501-504.

Goldman, Neal (1975) "Conceptual Generation", in Schank (ed.) **Conceptual Information Processing**, Elsevier.

Hovy, Eduard (1987) **Generating Natural Language under Pragmatic Constraints**, Ph.D. Dissertation, Yale University.

________ (1988) "Generating Language with a Phrasal Lexicon", in McDonald & Bolc (eds.) 1988, 353-384.

Jacobs, Paul (1988) "PHRED: a Generator for Natural Language Interfaces", in McDonald & Bolc (eds.) 1988, 312-354.

Kempen, Gerard (1987) editor, **Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics**, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Kukich, Karen (1988) "Fluency in Natural Language Reports", in McDonald & Bolc (eds.) 1988, 280-312.

Levelt, Willem J.M. & Herbert Schriffers (1987) "Stages of Lexical Access" in Kempen (ed.) 1987, 395-404.

Marcus, Mitchell (1987) "Generation Systems Should Choose Their Words", Proceedings of TINLAP-3, New Mexico State University, January 7-9, 211-214.

McDonald, David (1983) "Natural Language Generation as a Computational Problem: an introduction", in Brady & Berwick (eds.) **Computational Models of Discourse**, MIT Press, 1983, pp. 209-266.

_____, JoAnn Brooks, Beverly Woolf & Philip Werner, *Transition Networks for Discourse Management*, Technical Report #86-34, Dept. Computer & Information Science, UMass., April 1986.

_____ & Marie Meteer (1988) "From Water to Wine: generating natural language text from today's application programs", Proceedings of the 2d Conference on Applied Natural Language Processing (ACL), Austin, February 9-12, 41-48.

McGuire, Rod (1980) *Political Primaries and Words of Pain*, manuscript, Dept. of Computer Science, Yale; see also section 6 in Lehnert et al. "BORIS--An Experiment in In-Depth Understanding of Narratives" *Artificial Intelligence* **20** (1983) 15-62.

McKeown, Kathleen (1985) **Text Generation**, Cambridge University Press.

Meteer, Marie (1988) "Defining a Vocabulary for Text Planning", Proceedings of the Workshop on Text Planning and Realization, AAAI-88, St Paul, Minnesota, August 25, 115-112.

_____ (1989) *The SPOKESMAN Natural Language Generation System*, Technical Report 7090, Bolt, Beranek, & Newman Inc., Cambridge, MA.

_____, David McDonald Scott Anderson, David Forster, Linda Gay, Alison Huettner & Penelope Sibun (1987) **Mumble-86: Design and Implementation**, TR #87-87 Dept. Computer & Information Science, UMass., September 1987, 174 pgs.

Nirenburg, Sergei, Victor Lesser & Eric Nyberg (1989) "Controlling a Language Generation Planner", Proceedings of IJCAI-89, Detroit, August 21-25.

_____ & Irene Nirenburg (1988) "A Framework for Lexical Selection in Natural Language Generation", Proceedings of COLING-88, Budapest, August 22-27, 471-475.

Pustejovsky, James & Sergei Nirenburg (1987) "Lexical Selection in the Process of Language Generation", Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics, Stanford, July 6-9, 201-206.

_____ (1988) *Event Semantic Structure*, technical report, Brandeis University.

Rosenberg, Steve (1978) *Understanding in Incomplete Worlds*, AIM #475, Artificial Intelligence Lab, MIT.

Wilensky, Robert (1976) "Using Plans to Understand Natural Language", Proceedings of the annual meeting of the Association for Computing Machinery, Huston.