TELEGRAM: A GRAMMAR FORMALISM FOR LANGUAGE PLANNING

Douglas E. Appelt
Artificial Intelligence Center
SRI International
Menlo Park, California

0. Abstract

Planning provides the basis for a theory of language generation that considers the communicative goals of the speaker when producing utterances. One central problem in designing a system based on such a theory is specifying the requisite linguistic knowledge in a form that interfaces well with a planning system and allows for the encoding of discourse information. The TELEGRAM (TELEological GRAMmar) system described in this paper solves this problem by annotating a unification grammar with assertions about how grammatical choices are used to achieve various goals, and by enabling the planner to augment the functional description of an utterance as it is being unified. The control structures of the planner and the grammar unifier are then merged in a manner that makes it possible for general planning to be guided by unification of a particular functional description.

1. Introduction

By viewing language generation as a planning process, one can not only account for the way people use language to satisfy different goals they have in mind, but also model the broad interaction between a speaker's physical and linguistic actions. Formal models of planing can provide the basis for a theory of language generation in which communicative goals play a central role. Recent research in natural-language generation [1][2] has established the feasibility of regarding planning as the basis for the generation of utterances. This paper examines some of the problems involved in devising a grammar formalism for such a generation system that produces utterances and describes a particular implementation of a unification grammar, referred to as TELEGRAM, that solves some of these problems.

The KAMP system [1] was designed with the problems of multiple-goal satisfaction and the integration of physical and linguistic actions in mind. KAMP is a multiagent planning system that can be given a high-level description of an agent's goals, and then produce a plan that includes the performance of both physical and linguistic actions by several agents that will achieve the agent's goals. In the development of KAMP it was recognized that syntactic, semantic and pragmatic knowledge sources are necessary for the planning of utterances. These sources of knowledge were stored independently inside the system: a grammar was provided in addition to the axioms that constitute the agent's knowledge of the pragmatics of communication. However, rather than have one process that decides what to say, drawing on knowledge about the world and about communication, plus another independant process that decides how to encode that knowledge into English, KAMP employs a single process that uses both sources of knowledge to produce plans.

The primary focus of the research on KAMP was the representation and integration of the knowledge needed to make plans involving utterances. One area that was neglected was the representation of grammatical knowledge. KAMP relies on a very simple grammar composed of context-free rules that enable it to generate simple sentences. Such phenomena as gapping are totally outside of its capability. Because of the ad hoc nature of the representation, modifications and extensions of its linguistic coverage are very difficult.

Another criticism of KAMP's approach was that there was no obvious way to control the planning process. Instead of formulating a plan quickly, KAMP would search a large space of linguistic alternatives until it found an "optimal" solution. As some critics have pointed out, (e.g., [5]) such exhaustive planning is often not needed in practical situations — and is certainly not how people produce utterances in real time. KAMP would never produce an ungrammatical sentence, because it could always do unlimited backtracking after making an incorrect decision.

The remainder of this paper describes how to use a *unification grammar** to address these two problems of representation and control.

2. Unification Grammar

A unification grammar characterizes linguistic entities

This research was supported by the National Science Foundation under Grant MCS-8115105. The author is grateful to Barbara Grosz for helpful comments on earlier drafts of this paper.

* Unification grammar has often been referred to as *functional grammar* in the literature, e.g., [7], [11]. It is related to and shares many ideas with systemic grammar [6].

by collections of features called a *functional description* (*FDs*). Each of the features in an FD has a value that can be either atomic or another functional description. A unification grammar is a large FD that characterizes the features of every possible sentence in the language. In this paper, the FD that characterizes the intended utterance is called the *text FD* and the FD that constitutes the grammar is called the *grammar FD*.

The most salient feature of unification grammar that distinguishes it from other grammatical formalisms is its emphasis on linguistic *function*. All of the features used by the grammar have equal status, with functional and discourse-related features like *topic* and *focus* sharing equal status with grammatical roles like *subject* and *predicate*, and with syntactic categories like *NP* and *VP*.

Unification grammars are particularly well suited for language generation because they allow the encoding of discourse features in the grammar. A functional description can be constructed incorporating these features, and the syntactic details of the final utterance can then be specified through unification with the grammar FD. The process that constructs the text FD can treat it as a high-level blueprint fleshed out by unification, thereby relieving the high-level process of the need to consider low-level grammatical details. This strategy was used by McKeown [11].

Two functional descriptions can be *unified* by an algorithm that is similar to set union. Suppose F_1 and F_2 are functional descriptions. To compute the unification, F_3, of F_1 and F_2, written $F_3 = F_1 \bigoplus F_2$, the following algorithm is used:

If $\langle f_1, v_1 \rangle$ is a feature-value pair, and $\langle f_1, v_1 \rangle \in F_1$ and $\forall x \langle f_1, x \rangle \notin F_2$, or $\langle f_1, v_1 \rangle \in F_2$ and $\forall x \langle f_1, x \rangle \notin F_1$, then $\langle f_1, v_1 \rangle \in F_3$.

If $\langle f_1, v_1 \rangle \in F_1$ and $\langle f_1, v_2 \rangle \in F_2$, then $\langle f_1, v_3 \rangle \in F_3$, where the following conditions apply:

If $v_1 = \text{NIL}$ then $v_3 = v_2$, and similarly for v_2.

If $v_1 = \text{ANY}$ and $v_2 \neq \text{NONE}$, then $v_3 = v_2$, and similarly for v_2.

If $v_1 = v_2$, then $v_3 = v_1$.

If v_1 and v_2 are functional descriptions, then $v_3 = v_1 \bigoplus v_2$.

If any one of the above conditions fails, then the unification itself fails and the value of $F_1 \bigoplus F_2$ is undefined.

Functional descriptions can optionally contain a distinguished feature called *PATTERN* that is used to specify the surface order of constituents in the FD. The unification of two patterns is different in that it is based on deciding whether or not the orderings represented by the two patterns are consistent.

In spite of its advantages, there are some serious problems with unification grammar if it is employed straightforwardly in a language planning system. One of the most serious problems is the inefficiency of the unification algorithm as described above. A straightforward application of that algorithm is very expensive, consuming an order-of-magnitude more time in the unification process than in the entire planning process leading up to the construction of the text FD [11]. The problem is not simply one of efficiency of implementation. It is inherent in any algorithm that searches alternatives blindly and thereby does work that is exponentially related to the number of alternatives in the grammar. Any solution to the problem must be a conceptual one that minimizes the number of alternatives that ever have to be considered.

Another problem is that the text FD is not as high-level a blueprint as is really needed because every feature related to the speaker's intention to communicate must be part of the text FD when unification takes place. This implies, for example, that every descriptor that is part of a referring expression must be specified in advance. This may be unnecessary because for certain grammatical choices, the referring expression may be eliminated entirely. For example, in the by-phrase in a passive sentence, reference may be made pronominally (or not at all), in which case descriptors are unnecessary. Since the planner must know the linguistic context when planning descriptors, a noun-phrase FD is best constructed initially with a REFERENT feature, and later expanded by adding features that correspond to the descriptors.

While it is conceivable that the grammar could be designed to expand a REFERENT feature into a set of descriptors, that would amount to encoding in the grammar what is essentially a planning problem. This is undesirable because the grammar, being a repository of syntactic knowledge, should be separated from pragmatic knowledge. Conversely, it is also desireable to separate detailed syntactic knowledge from the planner, and the failure to do so was a major shortcoming with KAMP.

The next section describes how unification and planning can be combined to allow syntactic knowledge to be separated from the planner, but still allow the required flexibility of interaction between the planner and the grammar.

3. Combination of Unification and Planning

The TELEGRAM system solves the problems of efficiency and modularity through a close coupling between the processes of unification and planning. (The name TELEGRAM stands for TELEological GRAMmar because planning and goal satisfaction are integreated into the unification process.)

KAMP divided its actions into an abstraction hierarchy. The action hierarchy, as it pertains to linguistic actions,

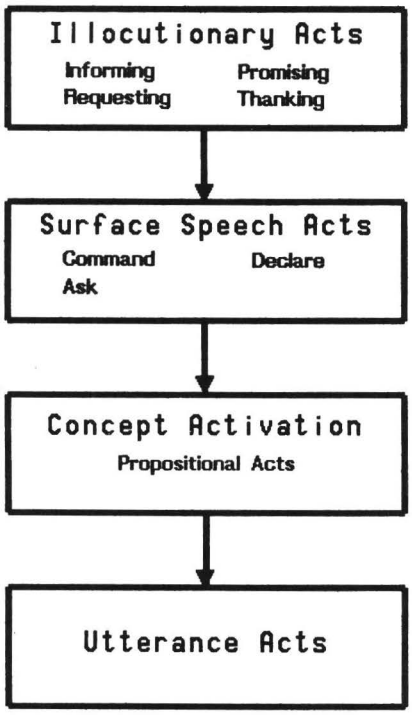

Figure 3.1
KAMP's Hierarchy of Linguistic Actions

is shown in Figure 3.1. Actions called illocutionary acts are at the top of the hierarchy, with surface speech acts and concept activation actions falling below, while the actual performance of the utterance is at the lowest level. Illocutionary acts are easily described at an abstract level that is best reasoned about by a conventional planning system, as was done in KAMP [1] and by Cohen [2]. However, as one progresses down the hierarchy, the planning becomes more and more dependent on the constraints of the grammar, although goal satisfaction is still very much a part of the reasoning that takes place. It is at the level of surface speech act and concept activation actions that the planning and unification processes can be most advantageously merged.

The means of combining planning and unification works as follows. At the time the planner plans to perform a surface speech act, enough information has been specified so that it knows the general syntactic structure of the sentence (declarative, interrogative, or imperative). A functional description of the utterance is created and then unified with the grammar.

This functional description is very general and does not contain sufficient information to specify a unique sentence. The functional description is elaborated *during the process of unification* so that it adds features incrementally to the functional description. The planner is called upon by the unification algorithm at the appropriate time to add the appropriate features. The end result is a functional description that is the same as if a complete functional description of the intended utterance had been unified with the grammar by means of a conventional unification algorithm that does not invoke planning.

The planner is invoked by the unifier when either of two situations arises:

- The unifier detects a feature in the text FD that has no corresponding feature in the grammar FD. Such features are a signal that elaboration must be performed. The feature is annotated with a goal wff that the planner plans to achieve, and the resulting actions specify additions to the functional description being unified. The unification process then continues in the normal manner.

- The unifier detects a choice in the grammar functional description that cannot be resolved through the unification of atomic features. Each choice in the grammar is annotated with a wff that describes to the planner what the effects of making the choice will be. The planner then decides which alternative is most consistent with its plans, making an arbitrary choice if insufficient information is available for a decision.

The combination of planning and unification that results has a number of benefits resulting from annotating a grammar with information useful to the planner, rather than trying to work linguistic knowledge into the planner in an ad hoc manner.

The ability to perform action subsumption, the opportunistic "piggybacking" of related goals as described in [1], is enhanced. Whether or not one can incorporate additional nonreferring descriptors into a noun phrase is governed by the structure and function of the noun phrase being planned. For example, a pronominal reference cannot incorporate any additional descriptors at all. Therefore, if a planner were to decide whether or not to perform action subsumption, it would have to know in advance how a referent was going to be realized. If this were to be performed before unification, the planner would have to have the detailed linguistic knowledge to know that it was possible. With a simple grammar like KAMP's this was possible, but with a larger grammar it is clearly undesirable.

The ability to do multiple-utterance and discourse planning is also enhanced. Since the grammar and planner are closely coupled, information can be easily fed back from the grammar to the planner. This feedback is one of the features that distinguish a language planning system from a system that first decides what to say, then how to say it. When an alternative is chosen, the planner has information about the goal that is to be achieved through the selection of that alternative. If unification based on that selection fails, the planner, instead of blindly trying other alternatives, can revise the entire plan — including

the incorporation of multiple utterances where only one was planned originally.

4. Example.

This example illustrates how a language system can use an annotated unification gramar like TELEGRAM. Suppose that there are two agents operating in an equipment assembly domain, and that the planning agent decides that the other agent should know that the location of a screwdriver $S1$ is in a particular toolbox, $TB1$. He then plans the illocutionary act**

$$\mathrm{Do}(\mathrm{AGT1}, \mathrm{Inform}(\mathrm{AGT2}, \mathrm{Location}(\mathrm{S1}) = \mathrm{TB1})).$$

The planner then plans a surface speech act consisting of a declarative sentence with the same propositional content as the illocutionary act. However, instead of constructing a syntactic-structure tree by using context-free rules, as in KAMP would do in this example, the TELEGRAM planner will create a high-level functional description of the intended utterance. For this example, the functional description would look like the following:***

$$\begin{bmatrix} \mathrm{CAT} = \mathrm{S} \\ \mathrm{SUBJ} = \begin{bmatrix} \mathrm{CAT} = \mathrm{NP} \\ \mathrm{REFERENT} = \mathrm{S1} \end{bmatrix} \\ \mathrm{VERB} = \begin{bmatrix} \mathrm{CAT} = \mathrm{V} \\ \mathrm{LEX} = \mathrm{BE} \end{bmatrix} \\ \mathrm{COMP} = \begin{bmatrix} \mathrm{CAT} = \mathrm{PP} \\ \mathrm{PREP} = [\mathrm{LEX} = \mathrm{IN}] \\ \mathrm{POBJ} = \begin{bmatrix} \mathrm{CAT} = \mathrm{NP} \\ \mathrm{REFERENT} = \mathrm{TB1} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

At this point, the planner is no longer directly in control of the planning process. The planner invokes the unifier with the above text functional description and the grammar functional description, and relinquishes control to the unification process.

The unification process follows the algorithm described in Section 2, until there is either an alternative in the grammar that needs to be selected or some feature in the text FD does not unify with any feature in the grammar FD.

In this example, the second of these situations arises when the noun phrase FD

$$\begin{bmatrix} \mathrm{CAT} = \mathrm{NP} \\ \mathrm{REFERENT} = \mathrm{TB1} \end{bmatrix}$$

** The precise meanings the elements of this representation are described in [1], but their intuitive meanings are adequate for understanding this paper.

*** Using the notation of Kay [7][8].

is unified with the functional description of a noun phrase from the grammar:

$$\begin{bmatrix} \mathrm{CAT} = \mathrm{NP} \\ \mathrm{PATTERN} = (\mathrm{DET\ MODS\ HEAD\ QUAL}) \\ \mathrm{DET} = [\cdots] \\ \mathrm{HEAD} = [\mathrm{CAT} = \mathrm{N}] \\ \mathrm{MODS} = [\cdots] \\ \mathrm{QUAL} = [\cdots] \end{bmatrix}$$

The FD for the noun phrase tells what the structure of the constituent is, but it does not contain a REFERENT feature. The straightforward application of the unification algorithm of Section 2 would simply yield the grammar FD along with the feature "REFERENT = TB1," which is not particularly useful. However, the feature REFERENT has an annotation that tells the unifier that the planner should be invoked with the goal of activating the concept TB1 for AGT2. The planner then plans a concept activation action, using its knowledge about AGT1 and AGT2's mutual knowledge, perhaps inserts a pointing action into the plan, and augments the text FD to resemble the following:

$$\begin{bmatrix} \mathrm{CAT} = \mathrm{NP} \\ \mathrm{DESC} = (\mathrm{Toolbox(TB1)}, \mathrm{Under(TB1, TABLE1)}) \end{bmatrix}$$

The new augmented functional description still does not unify with the grammar FD, but the annotation for the DESC feature is written to insert FDs corresponding to each of the descriptors into the text FD. This next expansion results in the following FD:

$$\begin{bmatrix} \mathrm{CAT} = \mathrm{NP} \\ \mathrm{DET} = \begin{bmatrix} \mathrm{NBR} = \mathrm{SG} \\ \mathrm{SUBCAT} = \mathrm{DEF} \end{bmatrix} \\ \mathrm{HEAD} = [\mathrm{LEX} = \mathrm{TOOLBOX}] \\ \mathrm{QUAL} = \begin{bmatrix} \mathrm{CAT} = \mathrm{PP} \\ \mathrm{PREP} = [\mathrm{LEX} = \mathrm{UNDER}] \\ \mathrm{POBJ} = \begin{bmatrix} \mathrm{CAT} = \mathrm{NP} \\ \mathrm{REFERENT} = \mathrm{TABLE1} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

This FD can be unified directly with the grammar FD, using the algorithm described in section 2. It is identical to the one that would have been planned had the entire FD been specified at the start of the unification process. However, by postponing some of the planning, and placing it under control of the unification process, the system preserves the ability to plan hierarchically while enhancing its ability to coordinate physical and linguistic actions.

5. Comparison with Related Systems.

There are several significant differences between TELEGRAM and other natural-language-generation systems that

have been developed using unification grammar or systemic grammar.

The TEXT system developed by McKeown [11] uses a unification grammar to generate coherent multisentential text and employs a straightforward unification algorithm. The unifier does not draw upon the system's pragmatic knowledge to decide among alternatives in the grammar, and being reduced to blind search, it requires a great deal of time to unify a single text functional description. The TEXT system does all its planning during the construction of the text FD and uses the unification process to fill in the grammatical details essential for producing the final utterance.

The NIGEL grammar designed by Mann [10] is a systemic grammar, but the philosophies underlying systemic and unification grammar are so similar that a comparison of the systems is warranted. The system "choosers" of NIGEL play a role similar to the annotations on the alternatives in TELEGRAM, and many other parallels can be drawn. The most fundamental difference between the two systems is in the assmptions underlying their design. NIGEL is intended to be completely independent of any particular application system or knowledge representation, an intention that has influenced all aspects of its design. A consequence of this decision is a complete separation of the grammatical processes from the other processes in the system, permitting communication only through a narrow channel. TELEGRAM, on the other hand closely couples reasoning about syntactic choices with the other planning done by the system, thereby enabling the reasoning about combined physical and linguistic actions. However, TELEGRAM sacrifices some of the simplicity of the interface between the grammar and the rest of the system.

6. Summary and Conclusion.

The TELEGRAM system described in this paper is an attempt to incorporate a large grammar into a language-planning system. This particular approach to representing knowledge in an annotated unification grammar and combining the processes of planning and unification results in the following advantages:

- Greater efficiency in the lower levels of the planning process, because the planner can be invoked to decide among alternatives, thus avoiding the reliance upon blind search.
- A simple method of resource allocation to the planning process by limiting the amount of backtracking the unifier is allowed to do.
- The ability to combine reasoning about physical and linguistic actions with a grammar that provides significantly wide coverage of the language.

Although the development of TELEGRAM is still in progress, early experience suggests that the TELEGRAM formalism has sufficient power to represent the syntactic knowledge of a language-planning system that efficiently encompasses a significant portion of English. A small grammar has been written that already has more power than the grammar of KAMP. Research is being conducted in discovering those discourse-related features that have to be included in a unification grammar. Although writing a "reversible" grammar does not appear to be feasible at this time, we hope this research will lead to the specification of a set of features that can be shared between unification grammars for parsing and for generation.

REFERENCES

[1] Appelt, Douglas E., *Planning Natural Language Utterances to Satisfy Multiple Goals,* SRI International Artificial Intelligence Center Technical Note No. 259, 1982.

[2] Cohen, Philip and C. R. Perrault, *Elements of a Plan-Based Theory of Speech Acts,* Cognitive Science, vol. 3, pp. 177–212, 1979.

[3] Cohen, Philip, *The Need for Identificaion as a Planned Action,* Proceedings of the Seventh International Joint Conference on Artificial Intelligence, 1981.

[4] Cohen, Philip, S. Fertig and K. Starr, *Dependencies of Discourse Structure on the Modality of Communication: Telephone vs. Teletype,* Proceedings of the Twentieth Annual Meeting of the Association for Computational Linguistics, 1982.

[5] Conklin, E. Jeffery, and D. McDonald, *Salience: The Key to the Selection Problem in Natural Language Generation,* Proceedings of the Twentieth Annual Meeting of the Association for Computational Linguistics, 1982.

[6] Halliday, M. A. K., **System and Function in Language,** Oxford University Press, London, 1976

[7] Kay, Martin, *Functional Grammar,* Proceedings of the Annual Meeting of the Linguistic Society of America, 1979.

[8] Kay, Martin, *Unification Grammar,* Xerox PARC tech report

[9] Kay, Martin, *An Algorithm for Compiling Parsing Tables from a Grammar*

[10] Mann, William C., and Christian Matthiessen, *Nigel: A Systemic Grammar for Text Generation,* University of Southern California Information Sciences Institute Technical Report ISI/RR-83-105, February, 1983.

[11] McKeown, Kathleen, *Generating Natural Language Text in Response to Questions about Database Structure,* Ph.D. dissertation, University of Pennsylvania, 1982.