DISCOURSE PRODUCTION

A computer model
of some aspects
of a speaker

ANTHONY DAVEY

Edinburgh University Press

Edinburgh University Press
22 George Square, Edinburgh
ISBN 0 85224 339 1
Printed in Great Britain by
Brown, Knight & Truscott Ltd,
Tonbridge, Kent

PREFACE

This book describes a computer program that produces English discourse. The program is capable of describing in a sequence of English sentences any game of noughts and crosses (tic-tac-toe), whether given or actually played with the program. The object is to understand something of what a speaker is doing when he speaks, and the program therefore demonstrates the operation of rules for selecting information into sentences, for connecting sentences into a discourse, and for constructing clauses, groups, and words to convey the required information with the maximum possible economy.

The program uses a systemic functional grammar to co-operate with semantic procedures in producing English. The grammar generates only a limited range of English, but one which is nonetheless sufficient to illustrate the advantages both theoretical and practical of such a grammar for a productive system.

Many other computer programs have accepted more or less natural English input, usually in the form of questions requiring an answer, but few have been designed to produce natural English, particularly connected discourse. As a producing system the present model offers a view of language use from a viewpoint slightly different from that of its predecessors. However, comprehension and production are dependent on each other, so that study of one may be expected to illuminate the other.

The work reported here was carried out between October 1970 and September 1973 at the Theoretical Psychology Unit of the School of Artificial Intelligence, Edinburgh University. It was financed by grants from the Science Research Council.

It is a pleasure to acknowledge the guidance and encouragement of Stephen Isard and Professor Christopher Longuet-Higgins who jointly supervised the project. The paper in its final form benefited greatly from Stephen Isard's criticism of earlier drafts, but it contains many shortcomings that could not properly be amended by a thesis supervisor, and for these I am entirely responsible.

CONTENTS

Preface vii

1. INTRODUCTION 1

2. GAME DESCRIPTIONS 8
   2.1 Introduction 8
   2.2 Planning discourse 8
   2.3 Sentence organisation 12
   2.4 Referring expressions 13
   2.5 Examples 17

3. A REVIEW OF PREVIOUS SYSTEMS 20
   3.1 Introduction 20
   3.2 The understanding hearer 20
   3.3 Local context 28
   3.4 Syntax 32

4. A SYSTEMIC FUNCTIONAL GRAMMAR 36
   4.1 Systemic grammar 36
   4.2 The scope of the program's grammar 42
   4.3 The [TYPE] system 42
   4.4 Clause systems 43
   4.5 Group systems 61
   4.6 Word systems 78
   4.7 [COORDINATION] systems 79

5. FUNCTIONS, AND THE REALISATION AND BUILDING RULES 84
   5.1 Introduction 84
   5.2 Functions 84
   5.3 Realisation relations 86
   5.4 Structure-building rules 88
   5.5 Clause rules 89
   5.6 Verb-groups 97
   5.7 Noun-group rules 97
   5.8 Preposition-group rules 102
   5.9 Adjective-group rules 104
   5.10 Doing without functions 104

6. A DESCRIPTION OF THE PROGRAM 109
6.1 A description of the program 109
6.2 An outline of the program 110
6.3 Designing a sentence 112
6.4 Introduction to the construction procedures 120
6.5 Making clauses 121
6.6 Making verb-groups 126
6.7 Making noun-groups 126
6.8 Making preposition-groups 144
6.9 Making adjective-groups 146
6.10 Making words 147

Appendix 1 150

Appendix 2 154

References 165

1
INTRODUCTION

A speaker must have an understanding of his audience. What he decides to convey and how he expresses it depend upon that understanding. In the simplest case, the speaker has information that he believes the audience lacks but would like to have. He gives this information in a monologue. This simple case is examined in the following chapters, with the aid of a computer program modelling the speaker. The program describes, in continuous English prose, any given game of noughts and crosses. Our objective is to show how a speaker gets from what needs to be said to the words that say it. The model therefore specifies how to decide what has to be put into words, how to divide this information into sentences, how to arrange a sentence so that its parts fit their context and are easy to understand, and then how to pick words and combine them into phrases to mean the right things. It also specifies, and this is perhaps the most interesting bit, what can be left unsaid: it attempts always to avoid telling the hearer anything he knows already, anything more than he needs to know, or anything he might reasonably be expected to work out for himself. Criteria for this are naturally somewhat arbitrary, but we shall find that the program is generally as tight lipped as possible consistent with saying things that are comprehensible.

First of all, then, the model defines a motive for discourse. The speaker's knowledge is compared with the knowledge that the hearer is presumed by the speaker to have of the subject matter, and the motive for discourse is to reduce any discrepancy revealed by the comparison. The speaker amends his assessment of the hearer's knowledge as the discourse progresses; in the present model, the speaker assumes that the hearer understands everything he is told, but a more sophisticated model would allow the hearer to ask questions. The discourse ends when the speaker believes that the discrepancy of knowledge has been resolved. The motive for the discourse, then, is that the speaker should get his model of what his hearer knows to correspond with the relevant parts of his own knowledge.

Having motivated the discourse in this way, we are forced to work out how to define the discrepancy of knowledge, taking account of what the hearer is presumed to have known initially and of what he is expected to infer. We are forced to structure the missing information, and to say how parts are taken from this structure into discourse units. Finally,

we are forced to say how a sentence and all its constituents are built to do particular jobs; we must say how referring expressions work, and in particular how determiners, modifiers, and pronouns are used. We must attend to the semantics of tense and aspect, and may occasionally use modal verbs.

At the highest level the model embodies a theory of how information within a small universe of discourse is selected and organised into a sequence of English sentences. Because the universe is so small, the task is simplified in three ways. It is easy to define what the speaker must convey to the hearer, namely a sequence of move descriptions; the speaker has only to see how far he has progressed through the game history to find exactly what more there is to say. In a larger context it would probably be much harder to formalise the assessment of the hearer's ignorance; of course, a natural way to assess someone's knowledge is to ask, and to permit him to ask, questions, and any development of the present model should include provision for interaction of speaker and hearer. In the second place, the task is simplified by the fact that the subject matter falls immediately into elements, the moves. Deciding how to structure the missing information just mentioned becomes much more difficult when the universe becomes even slightly richer. The third point is that the relation of each move element to its neighbour is fully defined in terms of the rules and point of the game. A richer context would require formalisation of a larger body of information about desires, expectations, and laws of nature; consider, for example, the different conjunctions in 'I planted roses, but greenfly destroyed them.' and 'I planted roses, and they flourished.' Charniak (1973) demonstrated how much such factors affect the way we tell a four-sentence story about a children's money-box, and we should expect to find comparable problems for any universe of discourse we might select.

The model incorporates a theory of grammar based originally upon the systemic grammar of Halliday (1961, 1967-8), but more immediately derived from Hudson (1971), which I shall call systemic functional grammar. It is a generative grammar, of a kind which has certain advantages for a language-producing system.

The grammar can be thought of as having two parts. The first is an analysis of the grammatical options open to any given item. A major clause item, for example, must be past or present tense but cannot have gender or number. This analysis is set out in a network of 'systems' in which each system is a set of simultaneous exclusive alternatives and the network structure exhibits the logical relation of each system to the rest. So past and present are the two options comprising the tense system, and the network is so constructed that a major clause item must be given a tense from the tense system, and, of course, cannot be given

gender or number. This part of the grammar is explained in Chapter 4.

The second part of the grammar comprises sets of rules, which state how the options open to an item may be constrained by the item's role in the grammatical environment. A simple example concerns case terminations. We know that a pronoun such as 'we' assumes the accusative form 'us' when dominated by a preposition, as in 'among us'. The grammar therefore contains a rule that constrains a pronoun in such an environment to have the accusative form. The rule is couched in terms of the pronoun's role, or 'function', in context. In turn the options selected for a particular item comprise a specification of the grammatical environment of the item's constituents, so further rules derive this environment from the selection made. All of these rules that relate from form and function are set out and explained in Chapter 5. They are presented in a formalism that is simple and easy to follow. The computer program in fact interprets these rules of the grammar as commands in a special language, but we shall not concern ourselves at all with the interpreter. The non-specialist will probably find it easier just to think of the rules being deployed as necessary than to follow the working of a computer procedure.

Systemic functional grammar has a certain practical advantage for the constructor of a language processing system. Being generative, it has complete and explicit rules of formation, which can be used to govern the production of grammatical items. The systems network sets out exactly what grammatical decisions must be taken in order adequately to characterise an item under construction, and the rules in Chapter 5 identify those decisions that are pre-empted by prior decisions about the composition of the grammatical environment. Decisions that are not pre-empted remain to be taken by program procedures that are semantic specialists. For example, the rules tell us that an item which is the object of a preposition must be accusative in form, if the accusative form is distinguished, but they do not say whether the item is to be singular or plural: that depends upon what we are trying to say and in particular upon what referent the item is to denote. The decision is therefore taken by a semantic specialist. The grammar, then, maps the campaign, distinguishing for each constituent those characteristics that are predetermined by the grammatical environment from those that have to be settled by reference to what the constituent must mean.

Another advantage of systemic functional grammar is that it seems to be psychologically more plausible than transformational grammar. Transformational grammarians have normally been cautious in expressing a view about the relation between a theory of grammar and the psychological processes of language use. However, it has been felt worthwhile to search for correlates in psycholinguistic behaviour of certain transformations (Bever 1971: 435, Kaplan 1971), and the

occurrence in many different languages of phenomena that can be given a common analysis within a transformational theory has been given a psycholinguistic significance (Bach & Harms 1968: 113). Within a particular language, idiolectal variation may be accounted for by referring to the varying depths at which a constraint upon a transformation applies, or to variations in the order of rule applications: such an explanation seems to be psycholinguistic as much as formal (Grinder & Postal 1971). It is therefore not unfair to mention shortcomings of transformational theory as the basis of a psychologically plausible model, and this we briefly do in section 4 of Chapter 5. Nonetheless, we shall be cautious in preferring the systemic functional grammar, asserting only that it enables the model to tackle problems that a speaker evidently tackles, and not that a speaker has a systemic grammar 'in his head'.

The present computer program, and presumably a speaker likewise, has information about the job to be done by the next utterance before it decides the form the utterance will assume to do it. The same can be said not only of utterances but also of each smaller constituent of the utterance. But systemic functional grammar rests upon an analysis of the functions performed by each grammatical item in its context, and states the relation between these functions and at least some of the grammatical characteristics of the item. It therefore corresponds well with the requirements of the productive model, and acquires a certain plausibility in consequence.

Systemic theory, unlike transformational grammar, does not confine itself within the bounds of a single surface-structure sentence. The functions of items are analysed within their context, and the scope of that context is to be as wide as is necessary for an adequate analysis. The grammar used in the program is in fact a very simple one, and so formalises only syntactic functions within the limits of a single surface structure independent clause. However, the boundaries of systemic analysis are being extended to include the pragmatic and social context of utterance (Fawcett 1973), and the program's grammar is in principle capable of extension to include these developments. A theory of grammar that accommodates the speaker's need to raise his eyes from the immediate sentence to the surrounding discourse is more plausible than one that doesn't.

We said a moment ago that determination of function precedes determination of form in the model's procedures. The same precedence should probably be true of discourse units larger than the surface sentence. We may recall demonstrations (Sachs 1967) that subjects normally forget the syntactic form, including sentence boundaries, of heard material very much quicker than they forget the meaning of it. It seems likely that in a similar way the speaker decides the information he

wishes to convey, or the social function to be performed, in his next piece of talk before he knows how many sentences he will divide his utterance into. There would in this case be some advantage in a theory of grammar such as the present one.

It is a commonplace that although Moliere's *gentilhomme* talked prose, he didn't invariably talk sentences. Completed utterances that are not well-formed sentences occur in a variety of circumstances. Others have investigated partial utterances in conversational exchanges, but the present model is not interactive and has nothing to say about these things. However, there is one type of ill-formed sentence which, although not produced by the current model, may be illuminated by it. As sentences of this sort are not uncommon in ordinary speech, there is an advantage in having a grammar that might accommodate the model's production of anomalies. A speaker sometimes fails to foresee that a particular part of his current sentence will be lengthy and complex and so will make the sentence unclear. This is particularly likely to happen in the construction of referring expressions, as the speaker realises the complexity of the modifiers and qualifiers needed to convey his intended referent. He may remedy the situation in one of a variety of ways, for example by inserting a parenthetical sentence: 'I met Jane's friend who – you know, you met her in Norfolk last year – and she said ...'.

Although the present program does not produce utterances like this, but instead produces somewhat elaborate referring expressions, the grammar upon which it is based would accommodate such anomalies in a simple and natural way, because the form of an item is specified as late as possible in the construction process. Whereas transformational grammar prefers to define all the transformations required for a surface sentence before applying any of them, systemic functional grammar determines the structure of each grammatical constituent only as its construction is taken in hand. It is therefore apparent that changes of plan, leading to anomalous utterances like the last example, are likely to cause only a local disturbance, confined to the part of the model responsible for making the constituent concerned. In the case of the last example, the part responsible for making the referring expression would break off and make the parenthetical sentence before reporting its task achieved: how it achieved it would not be predetermined or subject to review.

In this way the grammar would accommodate the production of anomalous sentences by delaying a decision about surface structure until it was inescapable. Planning great extents of surface structure before producing any of it not only implies an unlikely degree of prescience in the speaker, but also makes it harder to explain how anomalies appear at all. Systemic functional grammar, however, seems to provide a framework within which a model might produce anomalies

and for the right reasons. The grammar therefore gains further plausibility, although in this case the reason is not that the grammar enables the model to solve problems which the speaker evidently solves, but rather that it would enable the model, in a natural way, to fail to solve problems which the speaker evidently fails to solve.

To close this introduction, a word must be said about the reasons for casting the model in the form of a computer program. After all, the difficulty of writing a complex program compels the programmer to over-simplify and to take short cuts; he limits the range of choices open to the program, and makes simplifying assumptions. Such criticism is entirely justified, and throughout the following chapters, particularly Chapter 6, over-simplifications and assumptions will be mentioned. Nonetheless, a program may have the virtues of its vices. The programmer oversimplifies because the rules he is specifying for the computer to follow must be explicit, complete, and coherent. Even the very limited and simple grammar incorporated in the present model was improved from its original state by the computer's demands. The same argument applies, but with greater force, to the semantic specialist procedures. As we have seen, the objective is to show how discourse is constructed to convey information, and we must therefore state not only what grammatical options are open in any particular case, but also how choices are made between them. We must state not only that a noun group may be definite or indefinite, but also how a speaker decides which it should be. A verbal account of such a decision, lacking even the formalism of the rules of grammar, would be very liable to error, whereas a computer procedure can be tested not only for consistency but also for adequacy in varying circumstances.

But the most important reason for using a computer and a program to model the brain and mental processes is that a computer seems to be the most brain-like thing we have, and programs the closest analogy to the brain's processes. We need a procedural vocabulary to describe how a speaker gets from his intention to his utterance, and such vocabulary may be supplied from the theory and practice of programming. Winograd (1972) stressed the merits of his 'procedural grammar', particularly in connection with what he called 'demons', procedures responsible for dealing with coordinate conjunctions. His program was a team of semantic and syntactic specialist procedures any of which might take charge when called upon. This concept played an important part in Winograd's suggestions about psycholinguistic processes, and we should note that it derives from advances in programming practice. Another example is provided by Isard & Longuet-Higgins (1973): they made use of facilities provided by the POP-2 programming language to illuminate the relation between clause constituents, and in particular between the verb and nominal participants. Examples from the present program will

occur throughout the following chapters, but we might instance here the treatment of certain nominal clause participants. The programming language used allows us to treat procedures as passive objects, rather as the Queen of Hearts tried to use flamingoes. We can store information in a procedure, or let the procedure store information in itself, accessible to other procedures. We can put a prepared procedure on ice, and at a later time unfreeze it and let it run. This means that nominal participants can be moved into position, symbols in a symbol string, as though they were inert; but then any participant can be called upon to cooperate actively in the construction process. This capacity to be simultaneously a symbolic pigeon-hole and a procedure seems to be worth bearing in mind when we think about the brain's ability to perform computations.

We can, of course, use programming concepts without planning a program, and we can plan a program without writing or testing it. For example, a generative grammar is a program for operations upon symbol structures, but the majority of such grammars have never run as programs in a computer. Nonetheless, programmers know how hard it is to get a program right without testing it, and in particular to foresee correctly the interaction of the constituent procedures of a program such as the present one. The correction and development of a program not only results in a program that works, but may also stimulate new understanding. What the theory owes to the program, then, is likely to be simplicity, clarity, and a procedural language in which to express certain thoughts about psycholinguistic events.

2
GAME DESCRIPTIONS

2.1 INTRODUCTION

We come now to examine what the program does. The first three sections of this chapter illustrate respectively the arrangement of move descriptions into coherent discourse, the construction of sentences, and the construction of clause constituents, particularly referring expressions. The various points illustrated are then pulled together in examples of complete game commentaries, and a closing section explains the remarks made by the program when playing a game of noughts and crosses.

The examples given in this chapter have all been produced by the program. The program is written in the POP-2 language, and runs under the Multipop operating system on an Elliot 4130 computer. It needs 30K of storage in addition to space required by the Multipop system, and takes between thirty seconds and three minutes to produce each sentence, depending upon the complexity of the calculations required and the length of the final product.

2.2 PLANNING DISCOURSE

The program gives a commentary on a game, or part-game, of noughts and crosses. It assumes that the audience understands the game and follows the commentary as it is given. In order to help the audience, the program arranges the commentary in such a way that each separate sentence describes a coherent episode in the game, a move, or sequence of moves, which forms a 'play' in the struggle. The program has available certain sequential and contrastive conjunctions with which it signals to the audience the relation of one move to the next, and its deliberations about the arrangement of move-descriptions into sentences are influenced by a preference for making the fullest possible use of these signals. The program's resources include subordinating conjunctions, and so the program may at this stage decide not only what moves the next sentence will describe, but also whether a particular move will be described in a minor clause. We shall see later the circumstances that make this desirable. This part of the program is described in detail in Chapter 6, especially section 3.

The relation between one move and the next may simply be that of valid sequence, which the program conveys by 'and':

Move 1, and move 2. (i)
'You started the game by taking a corner,
and I took the opposite one.'
Move 1, move 2, and move 3. (ii)
'The game began with my taking a corner,
you took an adjacent one, and I took
the middle of the same edge.'

Such a run of moves related only by valid sequence is not very common in practice, and longer runs, if they occur, are always described in more than one sentence.

Contrastive conjunctions are more varied. A contrastive conjunction warns the hearer that something he had been led to expect didn't turn out that way. The conjunction links the expectation and disappointment into a chunk. A threat foiled is a natural example:

Move 1 but move 2. (iii)
'I threatened you by taking the middle
of the board but you blocked my line.'

There are, however, constraints upon sequences of contrastive conjunctions. It must always be immediately obvious what two pieces of information are being contrasted, and so the following example is confusing:

Move 1 but move 2 but move 3. (iv)
* 'I threatened you by taking the middle
of the board but you blocked my line
and threatened me but I blocked your
edge by taking the middle of it.'

The reason seems to be that the first 'but' leads the hearer to package the first move and the second move into a chunk, whereupon the second 'but' requires him to break that chunk and put moves two and three together instead. The difficulty is really that the second move has both a defensive and an agressive aspect; as Mr J.L. Stansfield has pointed out in conversation, an attractive solution is to mention these two aspects in two distinct sentences, as:

'I threatened you ... but you blocked (iva)
my line. That threatened me but I
blocked your edge.'

However, the present version of the system must complete its description of a move within a single sentence because of the way it keeps track of the progress of the commentary. It therefore represents this example by breaking the sentence after the second move. The sentence break makes a heavy pause, after which the third move in a sentence of its own can be contrasted with the whole preceding situation. In sentence-initial position 'but' is replaced by 'However':

Move 1 but move 2. However, move 3. (v)
'I threatened you by taking the middle of the board, but you blocked my line and threatened me. However, I blocked your edge by taking the middle of it.'

'However' contrasts the information in the sentence it introduces with the situation described by the preceding sentence. We therefore find it confusing if the sentence introduced by 'However' includes an internal contrast marked by but':

Move 1 but move 2. However, move 3 but move 4. (vi)
* 'I threatened you by taking the middle of the board, but you blocked my line and threatened me. However, I blocked your edge and threatened you but you blocked my diagonal and threatened me.'

'However' directs the hearer to contrast the whole of (move 3, move 4) with (move 1, move 2), whereas 'but' contrasts move 4 with move 3. As we have just seen, the system cannot produce

'I threatened you ... but you blocked my line and threatened me. However, I blocked your edge by taking the middle of it. That threatened you, but you blocked my diagonal and threatened me.' (via)

Since the system cannot make the contrasts of the move's two aspects separately, it drops the 'However' marking the contrastive link between the two sentences and produces

Move 1 but move 2. Move 3 but move 4. (vii)
'I threatened you ... but you blocked my line and threatened me. I blocked your edge and threatened you but you blocked my diagonal and threatened me.'

The system chooses to drop 'However' on the grounds that sentences are constructed to express the relation of their parts whereas relations between sentences are relatively secondary.

We said that the program signals a contrast only when it is clear what two items are contrasted. The program therefore does not produce

Move 1, and move 2 but move 3. (viii)
* 'I took the corner opposite the one I took first, you threatened me by taking the middle of the board but I blocked your diagonal.'

This is confusing, perhaps because the hearer is not immediately certain whether the third, blocking, move contrasts with both the preceding moves, or with just the most recent one. On the other hand, if the

contrasted pair of moves comes first in a run of three, no problem arises. The program may produce

Move 1 but move 2, and move 3. (ix)
'You threatened me by taking one of the
free corners but I blocked your edge,
and you forked me.'

The hearer is able to group moves 1 and 2 into a chunk, and then adjoins move 3 as an appendix. In fact such examples are rare, because the factors considered by the program normally dictate other arrangements. The present example is unusual both because it is taken from the description of an unfinished game in which the outcome of the fork is not known, and because the second, blocking, move was purely defensive.

In some cases where two moves within a sentence are contrasted, the speaker wants to warn the hearer not to raise his expectations too high upon hearing the first move of the pair. In such a case the first move may be described within a minor 'although' clause, which lets the hearer know in advance that what is about to be said didn't work out. For example, the tactical situation may be such that move 1, though defensive, cannot forestall defeat: the system avoids raising false hopes by producing

Although move 1, move 2. (x)
'Although you blocked one of my edges,
I won by completing the other.'

The system takes account of contrast in one other case. It draws attention to mistakes other than those made by itself, and contrasts the erroneous move with the better alternative:

Hypothetical but move 1. (xi)
'You could have forked me but you took
the square opposite the one you had just
taken.'

The modal verb warns the hearer, in advance of actually specifying the hypothetical move, that the move didn't really happen, and so not to expect too much of it. It seems equally natural to give the hearer an earlier warning by putting the hypothetical move in an 'although' clause instead of in a major clause followed by 'but':

'Although you could have forked me, you (xii)
took the square opposite the one you had
just taken.'

The system has the alternative of doing this, but only where the hypothetical move is decribed in just one simple clause. Where the description of the hypothetical move is more complicated, the system describes it in a major clause incorporating a conditional minor clause: