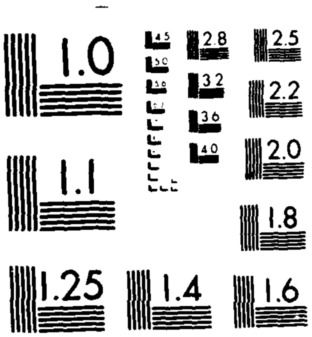


MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963 A

ISI/RR-82-102
September 1982

4

William C. Ma
Christian M.I. Matthiessen

Two Discourse Generators
A Grammar and a Lexicon for a Text-Production System

# Two Discourse Generators

William C. Mann

# A Grammar and a Lexicon for a Text-Production System

Christian M.I.M. Matthiessen

INFORMATION SCIENCES INSTITUTE

UNIVERSITY OF SOUTHERN CALIFORNIA ISI

*4676 Admiralty Way/Marina del Rey/California 90291*
*(213) 822-1511*

ISI/RR-82-102

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| | |
|---|---|
| **REPORT DOCUMENTATION PAGE** | READ INSTRUCTIONS BEFORE COMPLETING FORM |
| 1. AFOSR-TR- 83-0113 \| 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>Pa[illegible]. Two Discourse Generators<br>Part 2: A Grammar and a Lexicon for a Text-Production System | 5. TYPE OF REPORT & PERIOD COVERED<br>~~Research Report~~ Technica<br>6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>William C. Mann<br>Christian M. I. M. Matthiessen | 8. CONTRACT OR GRANT NUMBER(s)<br>F49620-79-C-0181 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>USC/Information Sciences Institute<br>4676 Admiralty Way<br>Marina del Rey, CA 90291 | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>61102F 2304/A2 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Mathematical and Information Sciences<br>Air Force Office of Scientific Research Building 410<br>Bolling Air Force Base Washington, D.C. 20332 | 12. REPORT DATE<br>September 1982<br>13. NUMBER OF PAGES<br>29 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)<br>.......... | 15. SECURITY CLASS. (of this report)<br>Unclassified<br>15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

This document is approved for public release distribution is unlimited.

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

..........

18. SUPPLEMENTARY NOTES

..........

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

(OVER)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

(OVER)

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19. KEY WORDS (continued)

Part 1: TWO DISCOURSE GENERATORS

computer generation of text, linguistically justified grammars, KDS, PROTEUS, natural language, systemic grammar, text generation

Part 2: ~~A GRAMMAR~~ AND A LEXICON FOR A TEXT-PRODUCTION SYSTEM

computer generation of text, linguistically justified grammars, knowledge representation methods, PENMAN, natural language, systemic grammar, text generation

20. ABSTRACT (continued)

Part 1: TWO DISCOURSE GENERATORS

Because discourse generation is a relatively new branch of Artificial Intelligence, only a few incomplete discourse generators have been developed. A study of those efforts reveals the nature of the task, what makes it difficult, and how the complexities of discourse generation can be controlled. This report compares two recent discourse generation systems: PROTEUS, by Anthony Davey at the University of Edinburgh, and KDS, by Mann and Moore at USC/Information Sciences Institute. Viewing the systems separately, the author identifies particular techniques in each system that contribute strongly to the quality of the resulting text. A comparison of the two systems follows, with a discussion of their common failings and the possibility of creating a new system that combines the strengths of both PROTEUS and KDS.

Part 2: A GRAMMAR AND A LEXICON FOR A TEXT-PRODUCTION SYSTEM

In a text-production system high and special demands are placed on the grammar and the lexicon. This report views these components in such a system. First, the subcomponents dealing with semantic and syntactic information are presented separately. The problems of relating these two types of information are then identified. Finally, strategies designed to meet the problems are proposed and discussed. One of the issues illustrated is what happens when a systemic linguistic approach is combined with a KL-ONE-like knowledge representation--a novel and hitherto unexplored combination.

ISI/RR-82-102
September 1982

# Two Discourse Generators

William C. Mann

# A Grammar and a Lexicon for a Text-Production System

Christian M.I.M. Matthiessen

INFORMATION SCIENCES INSTITUTE
UNIVERSITY OF SOUTHERN CALIFORNIA
4676 Admiralty Way/ Marina del Rey/ California 90291
(213) 822-1511

This research was supported by the Air Force Office of Scientific Research Contract No. F49620-79-C-0181. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research of the U.S. Government.

# PART 1: TWO DISCOURSE GENERATORS

William C. Mann

## Contents

# 1. WHAT IS DISCOURSE GENERATION?

The task of discourse generation is to produce multisentential text in natural language which (when heard or read) produces effects (informing, motivating, etc.) and impressions (conciseness, correctness, ease of reading, etc.) appropriate to a need or goal held by the creator of the text.

Because even little children can produce multisentential text, the task of discourse generation appears deceptively easy. It is actually extremely complex, in part because it usually involves many different kinds of knowledge. The skilled writer must know the subject matter, the beliefs of the reader, and his own reasons for writing. He must also know the syntax, semantics, inferential patterns, text structures, and words of the language. It would be complex enough if these were all independent bodies of knowledge, independently employed. Unfortunately, they are all interdependent in intricate ways. The use of each must be coordinated with all of the others.

For Artificial Intelligence, discourse generation is an unsolved problem. There have been only token efforts to date, and no one has addressed the whole problem. Still, those efforts reveal the nature of the task, what makes it difficult, and how the complexities can be controlled.

In comparing two AI discourse generators here we can do no more than suggest opportunities and attractive options for future exploration. We hope to convey the benefits of hindsight without too much detailed description of the individual systems. We describe them only in terms of a few of the techniques they employ, partly because these techniques seem more valuable than the system designs in which they happen to have been used.

# 2. THE TWO SYSTEMS

The systems we study here are PROTEUS, by Anthony Davey at Edinburgh [1], and KDS, by Mann and Moore at USC/Information Sciences Institute [6]. As we will see, each is severely limited and idiosyncratic in scope and technique. Comparison of their individual features reveals some technical opportunities.

Why do we study these systems rather than others? Both of them represent recent developments. Neither of them has the appearance of following a hand-drawn map or some other humanly produced sequential presentation. Thus their performance represents capabilities of the programs more than capabilities of the programmer. Also, they are relatively unfamiliar to the AI audience. Perhaps most important, they have written some of the best machine-produced discourse of the existing art.

First we identify particular techniques in each system which contribute strongly to the quality of the resulting text. Then we compare the two systems, discussing their common failings and the possibilities for creating a system having the best of both.

# 3. DAVEY'S PROTEUS

PROTEUS creates commentary on games of tic-tac-toe (noughts and crosses). Despite the apparent simplicity of this task, the possibilities of producing text are rich and diverse. (See the example in Appendix I.) The commentary is intended both to convey the game (except for insignificant variations of rotation and reflection) and to convey the significance of each move, including showing errors and missed opportunities.

PROTEUS can be construed as consisting of three principal processors, as shown in Figure 3-1.

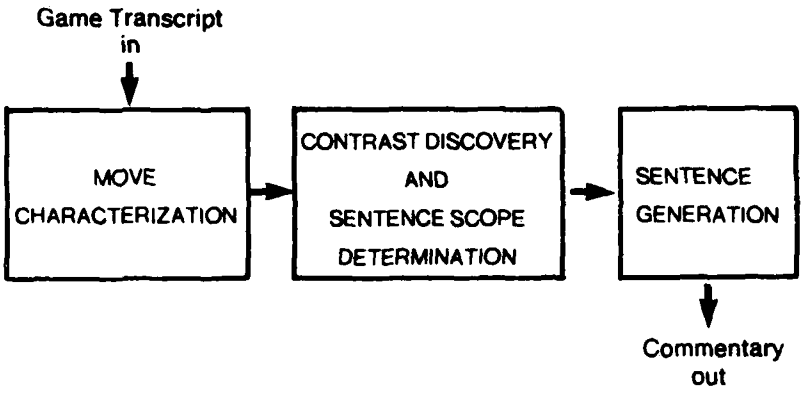


**Figure 3-1:** Principal processors of PROTEUS

**Move characterization** employs a ranked set of move generators, each identified as defensive or offensive, and each identified further with a named tactic such as blocking, forking, or completing a win. A move is characterized as being **a use of the tactic associated with the highest ranked move generator which can generate that move in the present situation.** The purpose of move characterization is to interpret the facts so that they become significant to the reader. (Implicitly, the system embodies a theory of the significance of facts.)

**Contrast** arises between certain time-adjacent moves and also between an actual move and alternative possibilities at the same point. For example,

- **Best move vs. Actual move:** The move generators are used to compute the "best" move, which is compared to the actual one. If the move generator for the best move has higher rank than any generator proposing the actual move, then the actual move is treated as a mistake, putting the best move and the actual move in contrast.

- **Threat vs. Block:** A threat contrasts with an immediately following block. This contrast is a fixed reflex of the system. It seems acceptable to mark any goal pursuit followed by blocking of the goal as contrastive.

Sentence scope is determined by several heuristic rules, including the following:

1. Express as many contrasts as possible explicitly (this leads to immediate selection of words such as "but" and "however").

2. Limit sentences to three clauses.

3. Put as many clauses in a sentence as possible.

4. Express only the worst of several mistakes.

The main clause structure is built before entering the grammar.

Both the move characterization process and the use of contrasts as the principal basis of sentence scope contribute a great deal to the quality of the resulting text. However, Davey's central concern was not with these two processes but with the third one, sentence generation. His system includes an elaborate systemic grammar, which he describes in detail in [1]. The grammar draws on the work of Halliday [2], Hudson [4], Winograd [12], Sinclair [8], Huddleston [3], and E. K. Brown, following Hudson most closely.[1]

Hudson's work offers a number of significant advantages to anyone considering implementing a discourse generation system:

1. Comprehensiveness. Its coverage of English is more extensive than comparable work.

2. Explicitness. The rules are spelled out fully in formal notation.

3. Unity. Since the grammar is defined in a single publication by one author, the issue of compatibility of parts is minimized.

It is interesting that Davey does not employ the systemic grammar derivation rules at the highest level. Although the grammar is defined in terms of the generation of sentences, Davey enters it at the clause level with a sentence description conforming to systemic grammar but built by other means. A sentence at this level is composed principally of clauses, but **the surface conjunctions have already been chosen.**

Although Davey makes no claim, this may represent a general result about text generation systems. Above some level of abstraction in the text planning process, planning is not conditioned by the content of the grammar. The obvious place to expect planning to become independent of the grammar is at the sentence level. But in both PROTEUS and KDS, operations independent of the grammar extend down to the level of independent clauses within sentences. Top-level conjunctions are not within such clauses, so they are determined by planning processes before the grammar is entered.

It would be extremely awkward to implement Davey's sentence scope heuristics in a systemic grammar. The formalism is not well suited for operations such as maximizing the total number of explicit contrastive elements. However, the problem is not just with the formalism; grammars generally do not deal with this sort of operation and are poorly equipped to do so.

[1] The direction of derivation in systemic grammar is the generation direction, so that there is no need to "reverse" it. Generation is divided into two kinds of activity: choices, which come in sets of alternatives (called "systems," hence "systemic") and rule applications. A system of choices (such as the choice between "demonstrative" and "possessive" among determiners which are "selective") is reached through other choices and so is conditional; but any choice, once reached, is unconstrained. Rule application is deterministic. Successive intervals of choice and rule application lead to a sequence of feature-sets, each of type "Word," which enable lexical substitutions. There are no transformations.

Although the computer scientist who tries to learn from Davey's *Discourse Production* [1] will find that it presents difficulties, the underlying system is interesting enough to be worth the trouble. Davey's implementation generally attempts to be orthodox, conforming to Hudson's *English Complex Sentences* [4]. Davey regularizes some of the rules toward type uniformity and thus reduces the apparent correspondence to Hudson's formulations. However, the linguistic base does not appear to have been compromised by the implementation.

One of the major strengths of Davey's work is that it takes advantage of a comprehensive, explicit, and linguistically justified grammar. Text quality is also enhanced by some simple filtering (of what will be expressed) based on dependencies between known facts. Some facts dominate others in the choice of what to say. If there is only one move on the board having a certain significance (say, "threat"), then the move is described by its significance alone, e.g., "you threatened me" without location information, since the reader can infer the locations. Similarly, only the most significant defensive and offensive aspects of a move are described even though all are known. The resulting text is diverse and of good quality. Although there are awkwardnesses, the immense advantage conferred by using a sophisticated grammar prevails.

## 4. MANN AND MOORE'S KDS

Space precludes a thorough description of KDS, but fuller descriptions are available [5], [6], [7].

KDS consists of five major modules, as indicated in Figure 4-2. A Fragmenter is responsible for extracting the relevant knowledge from the notation given to it and dividing that knowledge into small expressible units, which we call fragments or protosentences. A Problem Solver, a goal-pursuit engine in the AI tradition, is responsible for selecting the presentational style of the text and also for imposing the gross organization onto the text according to that style. A Knowledge Filter removes protosentences that need not be expressed because they would be redundant to the reader.

The largest and most interesting module is the Hill Climber, which has three responsibilities: to compose complex protosentences from simple ones; to judge relative quality among the units resulting from composition; and to repeatedly improve the set of protosentences on the basis of those judgments so that they are of the highest overall quality. Finally, a very simple Surface Sentence Maker creates the sentences of the final text out of protosentences. The data flow of these modules can be thought of as a simple pipeline, each module processing the relevant knowledge in turn.

The following are principal contributors to the quality of the output text:

1. The Fragment-and-Compose Paradigm. The information to be expressed is first broken down into an unorganized collection of subsentential (approximately clause-level) propositional fragments. Each fragment is created by methods which guarantee that it is expressible by a sentence (usually a very short one--this makes it possible to organize the remainder of the processing so that the text production problem is treated as an improvement problem rather than as a search for feasible solutions, a significant advantage). The fragments are then organized and combined in the remaining processing.

| KDS MODULES | MODULE RESPONSIBILITIES |
|---|---|
| FRAGMENTER | • Extraction of knowledge from external notation<br>• Division into expressible clauses |
| PROBLEM SOLVER | • Style selection<br>• Gross organization of text |
| KNOWLEDGE FILTER | • Cognitive redundancy removal |
| HILL CLIMBER | • Composition of concepts<br>• Sentence quality seeking |
| SURFACE SENTENCE MAKER | • Final text creation |

**Figure 4-2:** KDS module responsibilities

2. Aggregation Rules. Clause-combining patterns of English are represented in a distinct set of rules. The rules specify transactions on the set of propositional fragments and previous aggregation results. In each transaction several fragments are extracted and an aggregate structure (capable of representation as a sentence) is inserted. A representative rule, named "Common Cause," shows how to combine the facts for "Whenever C then X" and "Whenever C then Y" into "Whenever C then X and Y" at a propositional level.

3. Preference Assessment. Every propositional fragment or aggregate is scored using a set of scoring rules. The score represents a measure of sentence quality.

4. Hill Climbing. Aggregation and Preference Assessment are alternated under the control of a hill-climbing algorithm which seeks to maximize the overall quality of the collection, i.e., of the complete text. This allows a clean separation of the knowledge of what could be said from the choice of what should be said.

5. Knowledge Filtering. Propositions identified by an explicit model of the reader's knowledge as known to the reader are not expressed.

The knowledge domain of KDS' largest example is a Fire Crisis domain, the knowledge of what happens when there is a fire in a computer room. The task was to cause the reader, a computer operator, to know what to do in all contingencies of fire.

## 5. SYSTEM COMPARISONS

The most striking impression in comparing PROTEUS and KDS is that **they have very little in common**. In particular,

1. KDS has sentence scoring and a quality-based selection of *how to say things*; PROTEUS has no counterpart.

2. PROTEUS has a sophisticated grammar for which KDS has only a rudimentary counterpart.

3. PROTEUS has only a dynamic, redundancy-based knowledge filtering, whereas the filtering in KDS removes principally static, foreknown information.

4. KDS has clause-combining rules that make little use of conjunctions, whereas PROTEUS has no such rules but makes elaborate use of conjunctions.

5. KDS selects for brevity above all; PROTEUS selects for contrast above all.

6. PROTEUS takes great advantage of fact significance assessment, which KDS does not use.

The two systems have little in common technically, yet both produce high-quality text relative to predecessors. This raises an obvious question: Could the techniques of the two systems be combined in an even more effective system?

There is one prominent exception to this general lack of shared functic [illegible] a'd characteristics. Recent text synthesis systems [1], [6], [9], [10], [11] all include a facility for k[illegible]ing certain facts or ideas from being expressed. There is an implicit or explicit model of the rea[illegible] knowledge. Any knowledge which is somehow seen as obvious to the reader is suppressed. [illegible] f the implemented facilities of this sort are rudimentary; many consist only of *manually pr*[illegible] lists or marks. However, it is clear that they cover a deep intellectual problem. Discourse [illegible]ation must make differing uses of what the reader knows and what the reader does not know.

It is absolutely essential to avoid tedious statement of "the obvious." Proper use of presupposition (which has not yet been attempted computationally) likewise depends on this knowledge, and many of the techniques for maintaining coherence depend on it as well. But identification of what is obvious to a reader is a difficult and mostly unexplored problem. Clearly, inference is deeply involved, but what is "obvious" does not match what is validly inferable. It appears that as computer-generated texts become larger the need for a robust model of the obvious will increase rapidly.

## 6. POSSIBILITIES FOR SYNTHESIS

This section views the collection of techniques which have been discussed so far from the point of view of a designer of a future text synthesis system. What are the design constraints which affect the possibility of particular combinations of these techniques? What combinations are advantageous? Since each system represents a compatible collection of techniques, it is only necessary to examine compatibility of the techniques of one system within the framework of the other.

We begin by examining the hypothetical introduction of the KDS techniques of fragmentation, the explicit reader model, aggregation, preference scoring, and hill climbing into PROTEUS. We then examine the hypothetical introduction of PROTEUS' grammar, fact significance assessments, and contrast heuristic into KDS. Finally we consider the use of each system on the other's knowledge domain.

## 6.1. INTRODUCING KDS TECHNIQUES INTO PROTEUS

**Fragment-and-Compose** is clearly usable within PROTEUS, since the information on the sequence of moves, particular move locations, and the significance of each move all can be regarded as composed of many independent propositions (fragments of the whole structure). However, Fragment-and-Compose appears to give only small benefits, principally because the linear sequences of tic-tac-toe game transcripts give an acceptable organization and do not preclude many interesting texts.

**Aggregation** is also useable, and would appear to allow for a greater diversity of sentence forms than Davey's sequential assembly procedures allow. In KDS, and presumably in PROTEUS as well, aggregation rules can be used to make text brief. In effect, PROTEUS already has some aggregation, since the way its uses of conjunction shorten the text is similar to effects of aggregation rules in KDS.

**Preference Assessment** and **Hill Climbing** are interdependent in KDS. Introducing both into PROTEUS would appear to give great improvement, especially in avoiding the long awkward referring phrases PROTEUS produced. The system could detect excessively long constructs and give them lower scores, leading to the choice of shorter sentences in those cases.

The **Explicit Reader** model could also be used directly in PROTEUS; it would not help much, however, since relatively little foreknowledge is involved in any tic-tac-toe game commentary.

## 6.2. INTRODUCING PROTEUS TECHNIQUES INTO KDS

**Systemic grammar** could be introduced into KDS to great advantage. The KDS grammar was deliberately chosen to be rudimentary in order to facilitate exploration above the sentence level. (In fact, KDS could not be extended in any interesting way without upgrading its grammar.) Even with a systemic grammar in KDS, aggregation rules would remain, functioning as sentence design elements.

**Fact significance assessments** are also compatible with the KDS design. As in PROTEUS they would immediately follow acquisition of the basic propositions. They could improve the text significantly.

The **contrast heuristic** (and other PROTEUS heuristics) would fit well into KDS, not as an a priori sentence design device but as a basis for assigning preference. Higher scores for greater contrast would improve the text.

In summary, the principal techniques appear to be completely compatible, and the combination would surely produce better text than either system alone.

### 6.3. EXCHANGE OF KNOWLEDGE DOMAINS

The tic-tac-toe domain would fit easily into KDS, but the KDS text organization processes (not discussed in this paper) would have little to do. The fire crisis domain would be too complex for PROTEUS. It involves several actors at once, several parallel contingencies, and no single clear organizing principle. PROTEUS lacks the necessary text organization methods.

## 7. SHARED SHORTCOMINGS

These systems share (with many others) the primitive state of the art of computer-based discourse generation. Their processes are primarily devoted to activities that go without notice among literate people. The deeper linguistic and rhetorical phenomena usually associated with the term "discourse" are hardly touched. These systems make little attempt at coherence, and they do not respond in any way to the coherence (or lack of it) they achieve. Presupposition, topic, focus, theme, the proper role of inference, implicature, direct and indirect speech act performance, and a host of other relevant concepts all go unrepresented. Even worse, the underlying conceptual apparatus in both systems is extremely ad hoc and idiosyncratic, severely limiting the possiblities for using general knowledge of the semantics of English.

Despite these deficiencies, the systems produce relatively smooth, readable text. They are significant principally as collections of methods which may well survive to become parts of more competent and general-purpose systems of the future.

## 8. CONCLUSIONS

PROTEUS and KDS are distinct collections of discourse generation techniques. There is little overlap between them and no apparent conflict involved in using the techniques in combination. A system which combined the strengths of both would be significantly better than either.

# I. TEXT SAMPLES FROM THE TWO SYSTEMS

KDS: Fire-alarm text for computer operators

Whenever there is a fire, the alarm system is started, which sounds a bell and starts a timer. Ninety seconds after the timer starts, unless the alarm system is cancelled, the system calls Wells Fargo. When Wells Fargo is called, they, in turn, call the Fire Department.

When you hear the alarm bell or smell smoke, stop whatever you are doing, determine whether or not there is a fire, and decide whether to permit the alarm system or to cancel it. When you determine whether there is a fire, if there is, permit the alarm system, otherwise cancel it. When you permit the alarm system, call the Fire Department if possible, then evacuate. When you cancel the alarm system, if it is more than 90 seconds since the timer started, the system will have called Wells Fargo already, otherwise continue what you were doing.

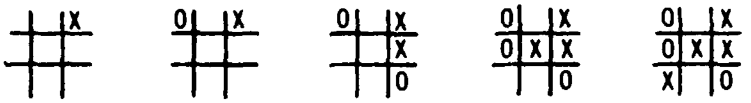
.......................................

PROTEUS: The move sequence 3169547

Position numbering

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

The moves:



The game started with my taking a corner, and you took an adjacent one. I threatened you by taking the middle of the edge opposite that and adjacent to the one which I had just taken but you blocked it and threatened me. I blocked your diagonal and forked you. If you had blocked mine, you would have forked me, but you took the middle of the edge opposite the corner which I took first and the one which you had just taken and so I won by completing my diagonal.

## REFERENCES

1. Davey, A., *Discourse Production*, Edinburgh University Press, Edinburgh, 1979.

2. Halliday, M. A. K. , *System ano Function in Language*, Oxford University Press, London, 1976.

3. Huddleston, R. D., *The Sentence in Written English: A Syntactic Study Based on an Analysis of Scientific Texts*, Cambridge University Press, London, 1971.

4. Hudson, R. A., *North-Holland Linguistic Series.* Volume 4: *English Complex Sentences*, North-Holland, London and Amsterdam, 1971.

5. Mann, W. C., and J. A. Moore, "Computer generation of multiparagraph English text," *American Journal of Computational Linguistics*, 1981.

6. Mann, W. C., and J. A. Moore, *Computer as Author--Results and Prospects*, USC/Information Sciences Institute, RR-79-82, 1980.

7. Moore, J. A., and W. C. Mann, "A snapshot of KDS, a knowledge delivery system," in *Proceedings of the Conference, 17th Annual Meeting of the Association for Computational Linguistics*, pp. 51-52, August 1979.

8. Sinclair, J. McH., A course in spoken English: Grammar, 1972.

9. Swartout, W. R., *A Digitalis Therapy Advisor with Explanations*, MIT Laboratory for Computer Science, Technical Report, February 1977.

10. Swartout, W. R., *Producing Explanations and Justifications of Expert Consulting Programs*, Massachusetts Institute of Technology, Technical Report MIT/LCS/TR-251, January 1981.

11. Weiner, J. L., "BLAH, a system which explains its reasoning," *Artificial Intelligence* 15, November 1980. 19-48.

12. Winograd, T., *Understanding Natural Language*, Academic Press, Edinburgh, 1972.

# PART 2: A GRAMMAR AND A LEXICON FOR A TEXT-PRODUCTION SYSTEM

Christian M. I. M. Matthiessen

## Contents

# ACKNOWLEDGMENTS

This research, the ideas stemming from it, and the substance of this report represent a joint effort. In particular I would like to thank William Mann, who has helped me think, given me many helpful ideas and suggestions, and commented extensively on drafts of the report; without him it would not be. I am also grateful to Yasutomo Fukumochi for helpful comments on a draft and to Michael Halliday, who has made clear to me many systemic principles and insights. Naturally, I am solely responsible for errors in the presentation and content.

# 1. THE PLACE OF A GRAMMAR AND A LEXICON IN PENMAN

This report will view a grammar and a lexicon as integral parts of a text-production system (PENMAN). This perspective reveals certain requirements for the form of the grammar and subparts of the lexicon, and leads to strategies for integrating these components with each other and with other parts of the system. These requirements will be addressed as the components, the subcomponents, and the integrating strategies are presented.

PENMAN, a successor to KDS [12, 13, 14], is being created to produce multisentential natural English text. It includes a knowledge domain encoded in a KL-ONE-like representation, a reader model, a text planner, a lexicon, and a sentence generator called NIGEL. NIGEL uses a systemic grammar of English of the type developed by Michael Halliday.

For present purposes the grammar, the lexicon, and their environment can be represented as shown in Figure 1-1.

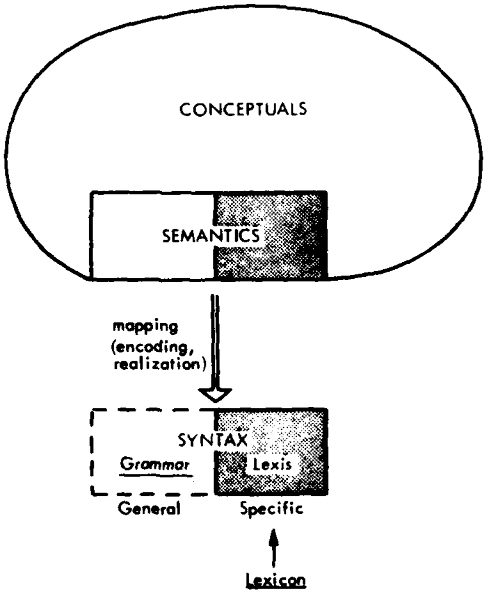


**Figure 1-1:** System overview

In Figure 1-1 the lines enclose sets and the boxes represent linguistic components. The dotted lines represent parts that have been developed independently and are being implemented, refined. and revised in this project. The continuous lines represent components being designed and developed within the project.

The box labeled *syntax* stands for syntactic information, both the general kind needed to generate structures (the grammar; the left part of the box) and the more specific kind needed for the syntactic definition of lexical items (the syntactic subentry of lexical items; to the right in the box--the term lexicogrammar can also be used to denote both ends of the box).

The other box (*semantics*) represents that part of semantics that has to do with our conceptualization of experience (distinct from the semantics of interaction--speech acts, etc., and the semantics of presentation--theme structure, the distinction between given and new information, etc.). In the figure, semantics is shown as one part of what is called *conceptuals*--our general conceptual organization of the world around us and our own inner world; it is the linguistic part of conceptuals. For the lexicon this means that lexical semantics is that part of conceptuals which has become lexicalized and thus enters into the structure of the vocabulary. There is also a correlation between conceptual organization and the organization of part of the grammar.

The double arrow between the two boxes represents the mapping (realization or encoding) of semantics into syntax. For example, the concept SELL is mapped onto the verb *sold*.[1]

*The grammar* is the general part of the syntactic box, the part concerned with syntactic structures. *The lexicon* cuts across three levels: It has a semantic part, a syntactic part (lexis), and an orthographic part (spelling, not present in the figure).[2] The lexicon consists entirely of independent lexical entries, each representing one lexical item (typically a word).

Figure 1-1, then, represents the part of the PENMAN text-production system that includes the grammar, the lexicon, and their immediate environment.

Because PENMAN is at the design stage, the discussion that follows is tentative and exploratory rather than definitive. The grammar is the most advanced component. It has been implemented in NIGEL, the sentence generator mentioned above. The grammar has been tested and is currently being revised and extended. None of the other components (those demarcated by continuous lines) has been implemented: they have been tested only by way of hand examples. This report will concentrate on the design features of the grammar rather than the results of its implementation and testing.

[1] I am using the general convention of capitalizing terms denoting semantic entries. Capitals will also be used for roles associated with concepts (like AGENT, RECIPIENT, and OBJECT) and for grammatical functions (like ACTOR, BENEFICIARY and GOAL). *These notions will be introduced below.*

[2] This means that an entry for a lexical item consists of three subentries: a semantic entry, a syntactic entry, and an orthographic entry. To emphasize the nature of the lexical entry, the lexicon box in Figure 1-1 (the shaded area) contains parts of both Syntax and Semantics

# 2. THE COMPONENTS

## 2.1. KNOWLEDGE REPRESENTATION AND SEMANTICS

### *Knowledge Representation*

One of the fundamental properties of the KL-ONE-like knowledge representation (KR) is its intensional/extensional distinction, the distinction between a general conceptual taxonomy and a second part of the representation (where we find individuals who can exist, states of affairs that may be true, etc.). This is roughly a distinction between what is conceptualizable and actual conceptualizations (whether real or hypothetical). In Figure 1-1, the two parts together are called conceptuals. As an example I will be using throughout this report, consider the intensional concept SELL, about which no existence or location in time is claimed. An intensional concept is related to extensional concepts by the Individuates relation: Intensiona' SELL is related to individual instances of extensional SELLs by the Individuates relation. If I know that Joan sold Arthur ice cream in the park, I have a SELL fixed in time which is part of an assertion about Joan: and the extensional SELL Individuates intensional SELL.[3] A concept has internal structure: it is a configuration of roles. The three roles associated with the concept SELL, that is, AGENT (the seller), RECIPIENT (the buyer) and OBJECT, form SELL's internal structure. These roles are like slots that are filled by other concepts. and the domains over which these concepts can vary are defined as value restrictions. The AGENT of SELL is a PERSON or a FRANCHISE, and so on. In other words, a concept is defined by its relation to other concepts (much as in European structuralism). These relations are roles associated with the concept, roles whose fillers are other concepts. This gives rise to a large conceptual net.

Another relation, SuperCategory, helps define the place of a concept in the conceptual net. SuperCategory gives the conceptual net a taxonomic (or hierarchic) structure in addition to the structure defined by the role relations. The concept SELL is defined by its place in the taxonomy by having TRANSACTION as a SuperCategory. If we want to, we can define a concept that will have SELL as a SuperCategory (i.e., bear the SuperCategory relation to SELL), for example, SELLOB 'sell on the black market'. As a result, part of the taxonomy of events is TRANSACTION---SELL---SELLOB.

If TRANSACTION has a set of roles associated with it, this set may be inherited by SELL and by SELLOB; this is a general feature of the SuperCategory relation. In the examples involving SELL that follow, I will concentrate on this concept and not try to generalize to its SuperCategories.

### *Semantic Subentry*

In Figure 1-1, semantics is shown as part of conceptuals; consequently, the set of semantic entries in the lexicon is a subset of the set of concepts. The subset is proper if we assume that there are concepts that have not been lexicalized (the assumption indicated in the figure). This assumption is perfectly reasonable: I have already introduced the concept SELLOB, for which there is no word in standard English. It is not surprising that we have formed concepts for which we have to create expressions rather than picking them ready-made from our lexicon. Furthermore, if we construct a conceptual component intended to support, say, a bilingual speaker, a number of concepts will be lexicalized in only one of the two languages.

[3] It should be emphasized that calling the concept SELL says nothing whatsoever about the English expression for that concept. the reasons for giving it this name are purely mnemonic The only way the concept can be associated with the word *sold* is through being part of a lexical entry.

A semantic entry, then, is a concept in the conceptuals. For *sold*, we find sold with its associated roles, AGENT, RECIPIENT, and OBJECT. The right-hand part of Figure 4-2 below (marked "se:", after a figure from Brachman's *A Structural Paradigm for Representing Knowledge* [1]), gives a more detailed semantic entry for *sold*: A pointer identifies the relevant part in the KR, the concept that constitutes the semantic entry (here the concept SELL).

The concept constituting the semantic entry of a lexical item has a fairly rich structure. Roles are associated with the concept and the modality (necessary or optional); the cardinality of and restrictions on (value of) the fillers are given.

Through the value restriction the linguistic notion of selection restriction is captured. *The stone sold a carnation to the little girl* is odd because the AGENT role of SELL is value restricted to PERSON or FRANCHISE, and the concept associated with *stone* falls into neither type.

The strategy of letting semantic entries be part of the knowledge representation would not have been possible in a notation designed to capture specific propositions only. However, since KL-ONE provides the distinction between intension and extension, the strategy is unproblematic in the present framework.

So what is the relationship between intensional/extensional and semantic entries? The working assumption is that for a large part of the vocabulary, the concepts of the intensional part of the KR may be lexicalized and thus serve as semantic entries. We have words for intensional objects, actions, and states, but not for individual extensional objects, etc., with the exception of proper names. They have extensional concepts as their semantic entries. For instance, *Alex* denotes a particular individuated person and *The War of the Roses* a particular individuated war.

Both the SuperCategory relation and the Individuates relation provide ways of walking around in the KR to find expressions for concepts. If we are in the extensional part of the KR, looking at a particular individual, we can follow the Individuates link up to an intensional concept. There may be a word for it, in which case the concept is part of a lexical entry. If there is no word for the concept, we will have to consider the various options the grammar gives us for forming an appropriate expression.

The general assumption is that all the intensional vocabulary can be used for extensional concepts in the way just described: Expressability is inherited with the Individuates relation.

Expression candidates for concepts can also be located along the SuperCategory link by going from one concept to another that is higher up in the taxonomy. Consider the following example: *Joan sold Arthur ice cream. The transaction took place in the park.* The SuperCategory link enables us to go from SELL to TRANSACTION, where we find the expression *transaction*.

### *Lexical Semantic Relations*

The structure of the vocabulary is parasitic on the conceptual structure. In other words, lexicalized concepts are related not only to one another, but also to concepts for which there is no word-encoding in English (i.e., nonlexicalized concepts).

Crudely, the semantic structure of the lexicon can be described as being part of the hierarchy of intensional concepts--the intensional concepts that happen to be lexicalized in English. The structure of English vocabulary is thus not the only principle reflected in the knowledge representation, but it is reflected. Very general concepts like OBJECT, THING, and ACTION are at the top of the hierarchy, and roles are inherited. This hierarchy corresponds to the semantic redundancy rules of a lexicon.

Considering the possibility of walking around in the KR and the integration of lexicalized and nonlexicalized concepts, the KR suggests itself as the natural place to state certain text-forming principles, some of which have been described under the terms Lexical Cohesion [8] and Thematic Progression [6].

I will now turn to the syntactic component in Figure 1-1, starting with a brief introduction to the framework (systemic linguistics) that does the same for that component as the notion of a semantic net did for the component just discussed.

## 2.2. LEXICOGRAMMAR

Systemic linguistics stems from a British tradition and has been developed by its founder, Michael Halliday [7, 9, 10], and other systemic linguists (see [4, 5] for a presentation of Fawcett's interesting work on developing a systemic model within a cognitive model) for over twenty years. Systemic linguistics covers many areas of linguistic concern, including studies of text, lexicogrammar, language development, and computational applications. Systemic grammar was used in SHRDLU [15] and more recently in another important contribution, Davey's PROTEUS [3].

The systemic tradition recognizes a fundamental principle in the organization of language: the distinction between *choice* and the *structures* that express (realize) choices. Choice is taken as primary and is given special recognition in the formalization of the systemic model of language. Consequently, a description is a specification of the choices a speaker can make together with statements about how he realizes a selection he has made. This realization of a set of choices is typically linear, for example, a string of words. Each choice point is formalized as a *system* (hence the name systemic). The options open to the speaker are two or more *features* that constitute alternatives which can be chosen. The preconditions for the choice are *entry conditions* to the system. Entry conditions are logical expressions whose elementary terms are features.

All but one of the systems have nonempty entry conditions. This causes an interdependency among the systems with the result that the grammar of English forms a network of systems that cluster when a feature in one system is (part of) the entry condition to another system. T s dependency gives the network depth: It starts (at its "root") with very general choices. Other systems of choice depend on this network (i.e., they have a feature from one of the systems--or a combination of features from more than one system--as an entry condition) so that the systems of choice become less general (more *delicate*, to use the systemic term) as we move along in the network.

The control of the nondeterministic part of the grammar resides in the network of systems. Systemic grammar thus contrasts with many other formalisms in that choice is given explicit representation and is captured in a single rule type (systems), not distributed over the grammar as, for example, optional rules of different types. This property of systemic grammar makes it a very useful component in a text-production system, especially in the interface with semantics and in ensuring accessibility of alternatives.

The rest of the grammar is deterministic--the consequence of features chosen from the network of systems. These consequences are formalized as feature *realization statements* whose task is to build the appropriate structure. For example, in independent indicative sentences, English offers a choice between declarative and interrogative sentences. If interrogative is chosen, this leads to a dependent system with a choice between wh-interrogative and yes/no-interrogative. When the latter is chosen, it is realized by having the FINITE verb before the SUBJECT.

Since the general design of the grammar is the focus of attention in this report, I will not go through the algorithm for generating a sentence as it has been implemented in NIGEL. The general observation is that the results are very encouraging, although incomplete. The algorithm correctly generates a wide range of English structures. There have been no serious problems in implementing a grammar written in the systemic notation.

Before turning to the lexico- part of lexicogrammar, I will give an example of the top-level structure of a sentence generated by the grammar. (I have left out the details of the internal structure of the constituents.)

| | | | | | |
|---|---|---|---|---|---|
| [1] | LOCATION | ACTOR | PROCESS | BENEFICIARY | GOAL |
| [2] | | SUBJECT | FINITE | | |
| [3] | THEME | | | | |
| | In the park | Joan | sold | Arthur | ice cream |

The structure consists of three layers of function symbols, all of which are needed to achieve the desired result. The structure is not only functional (with function symbols labeling the constituents instead of category names like Noun Phrase and Verb Phrase) but it is multifunctional.

Each layer of function symbols shows a particular perspective on the clause structure. Layer [1] gives the aspect of the sentence as a representation of our experience. The second layer structures the sentence as interaction between the speaker and the hearer: the fact that SUBJECT precedes FINITE signals that the speaker is giving the hearer information. Layer [3] represents a structuring of the clause as a message; the THEME is its starting point. The functions are called experiential, interpersonal, and textual, respectively, in the systemic framework: The function symbols are said to belong to three different metafunctions. In the rest of this report I will concentrate on the experiential metafunction, partly because it will turn out to be highly relevant to the lexicon.

### *Syntactic Subentry*

In the systemic tradition, the syntactic part of the lexicon is seen as a continuation of grammar (hence the term lexicogrammar for both of them): lexical choices are simply more detailed (delicate) than grammatical choices [9]. The vocabulary of English can be seen as one huge taxonomy, with *Roget's Thesaurus* as a very rough model.

A taxonomic organization of the relevant part of the vocabulary of English is intended for PENMAN, but this organization is part of the conceptual organization mentioned above. There is at present no separate lexical taxonomy.

The syntactic subentry will consist of two parts. There is always the class specification--the lexical features. This is a statement of the grammatical potential of the lexical item, that is, how it can be used grammatically. For *sold* the class specification is the following:

```
verb
class 10
class 02
benefactive
```

where "benefactive" says that *sold* can occur in a sentence with a BENEFICIARY, "class 10" that it encodes a material process (contrasting with *mental*, *verbal*, and *relational processes*), and "class 02" that it is a transitive verb.

In addition, there is a provision for a configurational part of the syntactic subentry, a fragment of a structure that the experiential part of the grammar can generate.[4] The structure corresponds to the top layer ( # [1]) in the example above. In reference to this example I can make more explicit what I mean by fragment. The general point is that (to take just one class as an example) the presence and character of functions like ACTOR, *BENEFICIARY, and* GOAL--*direct* participants in the event denoted by the verb--depend on the type of verb, whereas the more circumstantial functions like LOCATION remain unaffected and applicable to all types of verb. Consequently, the information about the possibility of having a LOCATION constituent is not the type of information that has to be stated for specific lexical items. The information given for them concerns only a fragment of the experiential functional structure.

The full syntactic entry for *sold* is as follows:

```
PROCESS = verb
          class 10
          ciass 02
          benefactive

ACTOR   =

GOAL    =

BENEFICIARY =
```

This example shows that *sold* can occur in a fragment of a structure in which it is a PROCESS and that there can be an ACTOR, a GOAL, and a BENEFICIARY. The usefulness of the structure fragment will be demonstrated in Section 4.

[4] This configurational part does not stem from the systemic tradition but is an exploration in the present design

# 3. THE PROBLEM

I will now turn to the fundamental problem of making a working system out of the parts that have been discussed. The problem has two parts:

1. the design of the system as a system with integrated parts, and
2. the implementation of the system.

I will be concerned with only the first aspect here.

The components of the system have been presented. What *remains--and that is the problem--is to design the missing links, to find the strategies that will connect those components.* Finding these strategies is a design problem in the following sense. The strategies do not come as accessories with the frameworks we have used (the systemic framework and the KL-ONE-inspired knowledge representation). Moreover, these two frameworks stem from two quite disparate traditions, with different sets of goals, symbols, and terms.

I will state the problem first for the grammar and then for the lexicon. As it has been presented, *the grammar* runs wild and free. It is organized around choice, to be sure, but there is nothing to relate the choices to the rest of the system, in particular to what we can take to be semantics. In other words, although the grammar may have a part that faces semantics--the system network, which (in Halliday's words) is semantically relevant grammar--it does not make direct contact with semantics. Additionally, if we know what we want the system to encode in a sentence, how can we indicate what goes where, that is, what a constituent (e.g., the ACTOR) should encode?

*The lexicon* incorporates the problem of finding an appropriate strategy to link the components to each other because it cuts across component boundaries. The semantic and syntactic subparts of a lexical entry have been outlined, but nothing has been said about how they should be matched with one another. This match is not perfectly straightforward partly because both entries may be structures (configurations) rather than single elements. In addition, there are lexical relations that have not yet been accounted for, especially synonymy and polysemy.

# 4. LOOKING FOR THE SOLUTIONS

## 4.1. THE GRAMMAR

### *Choice Experts and their Domains*

Control of the grammar resides in the network of systems. Choice experts can be developed to handle the choices in these systems. The idea is that there is an expert for each system in the network. This expert knows how to make a meaningful choice and what the factors influencing its choice are. It has at its disposal a table that tells it how to find the relevant pieces of information (somewhere in the knowledge domain, the text plan, or the reader model). In other words, the part of the grammar related to semantics is the part where the notion of choice is: The choice experts know about the semantic c ¬sequences of the various choices in the grammar and do the job of relating syntax to semantics.[5]

The recognition of different functional components of the grammar relates to the multifunctional character of a structure in systemic grammar mentioned in relation to the example in Section 2.2., *In the park Joan sold Arthur ice cream.* By organizing this sentence into PROCESS, ACTOR, BENEFICIARY, GOAL, and LOCATIVE, the grammar imposes an organization on our experience. The aspect of the sentence's organization relates to the conceptual organization of the knowledge domain: It is in terms of this organization (and not, for example, SUBJECT, OBJECT, THEME, and NEW INFORMATION) that the mapping between syntax and semantics can be stated. The functional diversity Halliday has provided for systemic grammar is useful in a text-production system.[6]

### *Pointers From Constituents*

For the choice experts to be able to work, they must know where to look. Assume that we are working on *in the park* in our example sentence *In the park Joan sold Arthur ice cream* and that an expert has to decide whether or not *park* should be definite. The information about the status in the reader's mind of the concept corresponding to *park* in this sentence is located at this concept: The trick is to associate the concept with the constituent being built. In the example structure given earlier, *in the park* is both LOCATION and THEME, only the former of which is relevant to the present problem. The solution is to set a pointer to the relevant extensional concept when the function symbol LOCATION is inserted, so that LOCATION will carry the pointer and thus make the information attached to the concept accessible.

## 4.2. THE LEXICON AND THE LEXICAL ENTRY

I have already introduced the semantic subentry and the syntactic subentry. They are stated in a KL-ONE-like representation and a systemic notation, respectively. The question now is how to relate the two.

In the knowledge representation the internal structure of a concept is a configuration of roles. These roles lead to new concepts to which the original concept is related. A syntactic structure is seen as a configuration of function symbols. Syntactic categories serve these functions; in the generation of a structure the functions lead to an entry of a part of the network. For example, the function ACTOR leads to a part of the network whose entry feature is Nominal Group, just as the role AGENT (of SELL) leads to the concept that fills it. The parallel between the two representations in this area is the following:

[5] A possible definition of the full semantics of the grammar is, as a result of this approach, "semantics = what the grammatical choice experts look at." In the present discussion, I have focused on the knowledge domain only, partly because this is the area most relevant to lexical semantics.

[6] Space limitations prohibit a discussion in this report of other functions in Halliday's systemic grammar.

| KNOWLEDGE REPRESENTATION | SYNTACTIC REPRESENTATION |
|---|---|
| role | function |
| filler | exponent |

where *exponent* denotes the entry feature into a part of the network (e.g., Nominal Group) that the function leads to.

This parallel clears the path for a strategy for relating the semantic entry and the syntactic entry. The strategy is in keeping with current ideas in linguistics.[7] Consider the following crude entry for *sold*, given here as an illustration:

Subentries:

| semantic | | syntactic | | | orthographic |
|---|---|---|---|---|---|
| | | Functions | | Lexical features | |
| SELL-concept | = | PROCESS | = | verb<br>class 10<br>class 02<br>benefactive | "sold" |
| AGENT | = | ACTOR | | | |
| OBJECT | = | GOAL | | | |
| RECIPIENT | = | BENEFICIARY | | | |

where the previously discussed semantic and syntactic subentries are repeated and paired off against each other

This full lexical entry makes clear the usefulness of the second part of the syntactic entry--the fragment of the experiential functional structure in which *sold* can be the PROCESS.

Now another piece of the total picture falls into place. The notion of a pointer from an experiential function (like BENEFICIARY) in the grammatical structure to a point in the conceptual net was introduced above. We can now see how this pointer may be set for individual lexical items: It is introduced as a simple relation between a grammatical function symbol and a conceptual role in the

[7] The mechanism for mapping has much in common with one developed for Lexical Functional Grammar (see, for example [2]), although the levels are not the same. The entry also resembles a lexical entry in the Pan-Lexicalism framework developed by Hudson [11]

lexical entry of, for example, SELL. Since there is an Individuates link between this intensional concept and any extensional SELL, the extensional concept that is part of the particular proposition being encoded grammatically, the pointer is inherited and will point to a role in the extensional part of the knowledge domain.

At this point I will refer again to the Figure 4-2, whose right half I have already referred to as a full example of a semantic subentry ("se:"). "sp:" is the spelling or orthographic subentry; "ge:" is the syntactic subentry.

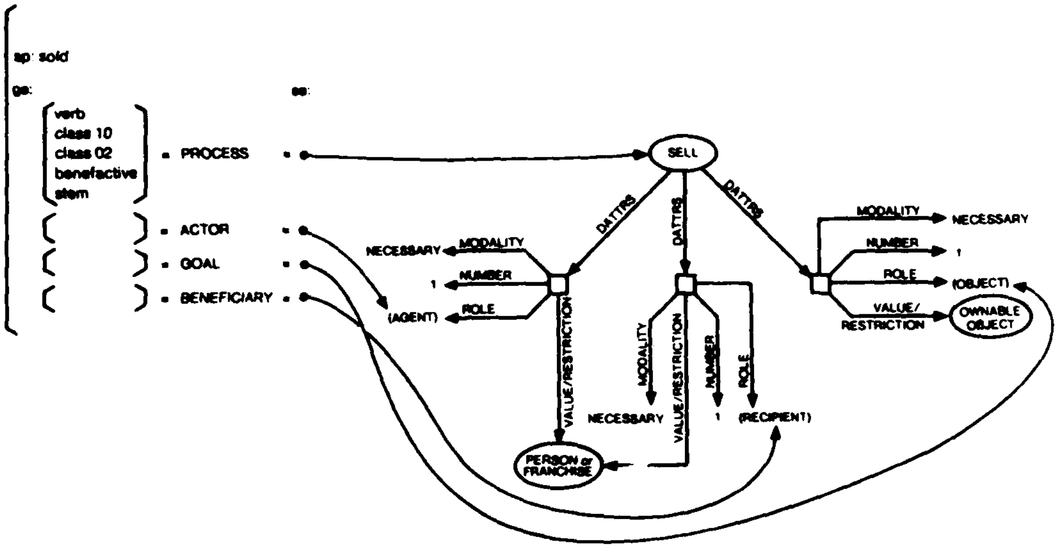


**Figure 4-2:** Lexical entry for *sold*

We have two configurations in the lexical entry: in the semantic subentry, the concept plus a number of roles; in the syntactic subentry, a number of grammatical functions. The match is represented in Figure 4-2 by the arrows.

All three roles of SELL have the modality "necessary." This does not dictate the grammatical possibilities. The grammar in NIGEL offers a choice between, for example, *They sold many books to their customers* and *The book sold well*. In the second example, the grammar only picks out a subset of the roles of SELL for expression. In other words, the grammar makes possible the adoption of different perspectives.[8] I can now return to the observation that the functional diversity Halliday has provided for systemic grammar is useful for our purposes. The fact that grammatical structure is multilayered means that those aspects of grammatical structure relevant to the mapping between the two lexical entries are identified, made explicit (as ACTOR, BENEFICIARY, etc.), and kept separate from principles of grammatical structuring that are not directly relevant to this mapping (e.g., SUBJECT, NEW, and THEME).

[8] The strategy of letting the functional syntactic entry pick up different parts of a concept and adopt different perspectives finds many uses, e.g., in the treatment of pairs like *buy* vs. *sell* and *give* vs. *receive* and in the account for nominalizations

In conclusion, a strategy for accounting for *synonymy* and *polysemy* can be mentioned. The way to capture synonymy is to allow a concept to be the semantic subentry for two distinct orthographic entries. If the items are syntactically identical as well, they will also share a syntactic subentry. Polysemy works the other way: There may be more than one concept for the same syntactic subentry.

## 5. CONCLUSION

I have discussed a grammar and a lexicon for PENMAN in two steps. First I looked at them as independent components--the semantic entry, the grammar, and the syntactic entry--and then, after identifying the problems of integrating them into a system, I turned to strategies for relating the grammar to the conceptual representation and the syntactic entry to the semantic entry within the lexicon.

In the first step I introduced the KL-ONE-like knowledge representation and the systemic notation and indicated how their design features can be put to good use in PENMAN. For instance. the distinction between intension and extension in the knowledge representation makes it possible to let lexical semantics be part of the conceptuals. I also suggested that the relations SuperCategory and Individuates can be used to find expressions for a particular concept.

The second step attempted to connect the grammar to semantics through the notion of the choice expert, making use of a design principle of systemic grammars where the notion of choice is taken as basic. I pointed out the correlation between the structure of a concept and the notion of structure in the systemic framework and showed how the two can be matched in a lexical entry and in the generation of a sentence, a strategy that could be adopted because of the multifunctional nature of structure in systemic grammars. This second step has been at the same time an attempt to start exploring the potential of a combination of a KL-ONE-like representation and a systemic grammar.

Although many aspects have had to be left out of the discussion, a number of issues are of linguistic interest and significance. The most basic one is perhaps the task itself: designing a model where a grammar and a lexicon can actually be made to function as more than just structure generators. One issue related to this is that various parts external to the grammar find resonance in different parts of the grammar, and there is a partial correlation between the conceptual structure of the knowledge representation and the grammar and lexicon.

As was emphasized in the introduction, PENMAN is at the design stage: There is a working sentence generator, but the other aspects of what has been discussed have not been implemented. and there is no commitment yet to a frozen design. Naturally, a large number of problems still await their solution, even at the level of design; clearly, many of them will have to wait. For example, selectivity among terms, beyond referential adequacy, is not addressed.

In general, while noting correlations between linguistic organization and conceptual organization, we do not want the relation to be deterministic: Part of being a good verbalizer is being able to adopt different viewpoints, to verbalize the same knowledge in different ways. This is clearly an area for future research. Ideas such as grammars organized around choice and choice experts should prove useful tools in working out extensions.

# REFERENCES

1. Brachman, R., *A Structural Paradigm for Representing Knowledge*. Bolt Beranek and Newman Inc., Technical Report, 1978.

2. Bresnan, J., "Polyadicity: Part I of a theory of lexical rules and representation," in T. Hoekstra H. van der Hulst, and M. Moortgat (eds.), *Lexical Grammar*, Dordrecht, 1980.

3. Davey, A., *Discourse Production*, Edinburgh University Press, Edinburgh, 1979.

4. Fawcett, R. P., *Exeter Linguistic Studies*. Volume 3: *Cognitive Linguistics and Social Interaction*. Julius Groos Verlag Heidelberg and Exeter University, 1980.

5. Fawcett, R. P., Systemic functional grammar in a cognitive model of language. University College, London. Mimeo, 1973.

6. Danes, F., ed., *Papers on Functional Sentence Perspective*. Academia, Publishing House of the Czechoslovak Academy of Sciences, 1974.

7. Halliday, M. A. K., "Categories of the theory of grammar," *Word 17*, 1961.

8. Halliday M. A. K. and R. Hasan, *Cohesion in English*, Longman, London, 1976. English Language Series, Title No. 9.

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

10. Hudson, R. A., *North-Holland Linguistic Series*. Volume 4: *English Complex Sentences*. North-Holland, London and Amsterdam, 1971.

11. Hudson, R. A., DDG working papers. University College, London. Mimeo.

12. Mann, W. C., and J. A. Moore, "Computer generation of multiparagraph English text," *American Journal of Computational Linguistics*, 1981.

13. Mann, W. C., and J. A. Moore, *Computer as Author--Results and Prospects*. USC/Information Sciences Institute, RR-79-82, 1980.

14. Moore, J. A., and W. C. Mann, "A snapshot of KDS, a knowledge delivery system," in *Proceedings of the Conference, 17th Annual Meeting of the Association for Computational Linguistics*, pp. 51-52, August 1979.

15. Winograd, T., *Understanding Natural Language*, Academic Press, Edinburgh, 1972.

LME

8