

CONSTRAINTS ON THE GENERATION OF ADJUNCT CLAUSES

Alison K. Huettner * Marie M. Vaughan ** David D. McDonald **

Department of Linguistics *
Department of Computer & Information Science **
University of Massachusetts
Amherst, Massachusetts 01003

ABSTRACT

This paper presents an analysis of a family of particular English constructions, all of which roughly express "purpose". In particular we look at the purpose clause, rationale clause, and infinitival relative clause. We (1) show that couching the analysis in a computational framework, specifically generation, provides a more satisfying account than analyses based strictly on descriptive linguistics, (2) describe an implementation of our analysis in the natural language generation system MUMBLE-86, and (3) discuss how our architecture improves upon the techniques used by other generation systems for handling these and other adjunct constructions.

1. INTRODUCTION

Natural language provides a variety of devices for expressing relations between elements in a text. Simply positioning two sentences in sequence conveys an implicit relation between them:

(1) *I bought a book. I'm going to read it on the plane.*

Clauses may also be joined with explicit lexical connectives:

(2) *I bought a book so that I could read it on the plane.*

A few relations may be expressed directly through particular types of subordination of one clause to another¹:

(3) *I bought a book to read on the plane.*

This latter category is the most cohesive of these three devices, as the adjunct is crucially dependent on the material in the matrix clause for its interpretation (Halliday & Hasan, 1976). However, such structural linking mechanisms are also the most limited in applicability: only certain relations may be expressed

in this way and complex grammatical constraints must be satisfied.

In this paper, we analyze a particular class of structural devices, including the purpose clause (exemplified in 3 above), rationale clause, and infinitival relative, from the perspective of natural language generation. All three constructions express kinds of "purpose": purpose clauses express the use to which someone will put an object that is expressed in the main clause; rationale clauses express the overall intention behind the main clause action; infinitival relatives express the usual function of their NP head.² We look at what underlying semantic relations license the constructions, the constraints on the syntactic form of the main and adjunct clauses, and the gapping pattern of the arguments of each adjunct. We discuss these as information needed by the generator in order for it to choose and use these devices correctly and discuss at what stages in the generation process the information must be applied.

We contrast our analysis with those typically given from the perspective of generative-transformational linguistics, particularly thematic analyses, concluding that an analysis that considers the construction in a particular situation and in terms of a coherent model of the world can capture the constraints more easily. We provide a particular example implemented in the natural language generation system MUMBLE-86 (McDonald, 1984) and show how our analysis may be generalized to similar structural adjunct constructions. We further show that many earlier approaches to generating complex sentences (Derr & McKeown, 1984; Davey, 1974; Kukich, 1985; Mann & Moore, 1981) have architectural limitations that would keep them from handling these types of constructions with any generality.

¹ We refer here to infinitive clauses which are grammatically related to the main clause as optional adverbials rather than as complements (arguments) to a verb, such as "*Floyd wanted to go to the zoo*".

² The notion of "purpose" is of course ambiguous between "intention" and "function".

2. DESCRIPTION OF THE CONSTRUCTION

Before addressing the generation of adjunct infinitive clauses, it is necessary to define our terms and distinguish the different constructions. We will begin by discussing purpose clauses³ and then contrast them with rationale and infinitival relative clauses.

2.1 Purpose clauses

A purpose clause (PC) expresses the purpose or intended use of a particular object which the main clause is in some sense "about". It is attached as a daughter of VP and is fixed in VP final position. It has the following variants, distinguished trivially by the position of the gap:⁴

- (4) a. *I bought the shelf; [e_i to hold my cookbooks]*
- b. *I bought the cookies; [for Mary to eat e_j]*
- c. *I bought the cushion; [for Mary to sit on e_j]*

The sentences in (4) demonstrate that PC has one obligatory gap, which can occur in any of its NP argument positions: subject position, as in (a);⁵ direct object position, as in (b); or prepositional object position, as in (c). The gap is coreferential with ("controlled by") the direct object of an SVO main clause, or with the subject of a passive or unaccusative main clause. This pattern of antecedents has been variously characterized as deep structure (direct) objects (Huettner, 1987; implicitly in Rappaport & Levin, 1986); as arguments bearing the thematic role of Theme (Faraci, 1974; Williams, 1980); or as entities whose availability for further manipulation plays a part in the semantics of the sentence (Jones, 1985).

A PC with its obligatory gap in non-subject position (like those in 4b,c above) may have an additional subject gap, as shown in (5):

- (5) *I_j bought it; [e_j to eat e_i]*

This second gap is optional, and the determination of its antecedent is more complex than the controller of the obligatory gap. In (5) the PC subject is coindexed with the matrix (main clause) subject; however, (6) shows that an indirect object takes

³ The purpose clause has also been known as a "retroactive purpose clause", for example in Jespersen (1940). Jespersen reserves the term "purpose clause" for what we are calling a rationale clause; however, our terminology dates at least from Faraci (1974) and is used by Bach (1982) and Jones (1985) among others.

⁴ The symbol "e" stands for an empty category, or gap, in an argument position. The subscripts indicate coreference.

⁵ A PC with a subject gap is often called an objective clause.

precedence over the subject as controller for this gap:⁶

- (6) a. *I gave it; to Mary_k [e_k to read e_i]*
- b. **I_j gave it; to Mary_k [e_j to read e_i]*

When there is no suitable antecedent in the matrix, the optional subject gap will have arbitrary or indefinite reference:

- (7) a. *This box; was purchased [e_{arb} to keep supplies in e_i]*
- b. *These doughnuts; are [e_{arb} to eat e_i]*

The set of antecedents for the optional subject gap has been characterized configurationally, as the "closest" NP argument after the obligatory gap has found an antecedent (Chomsky, 1980); thematically, as the highest NP argument on a "thematic hierarchy" ranging from Goal to "arbitrary" (Nishiguchi, 1984); and pragmatically, as the person in whose control the Theme is at the time of the action (Ladusaw & Dowty, 1985).

2.2 Rationale Clause

Easily confused with the purpose clause is the rationale clause (RatC), also known as an "in order to" clause or result clause. RatC can be distinguished from PC by the fact that RatC permit only subject gaps, whose antecedent is (usually) the matrix subject, rather than its object. Note the ambiguity of the following:

- (8) *Amy Lou; took Mildred; to the zoo e_i; to feed the lions.*

On the PC reading, Mildred is feeding the lions; on the RatC reading Amy Lou is feeding the lions (possibly using Mildred as lion food). A RatC reading may always be paraphrased with *in order*, as in (9), to rule out the PC reading:

- (9) *Amy Lou ; took Mildred; to the zoo in order e_i; *j; to feed the lions.*

In contrast with PC, the controller of a RatC gap need not be any argument of the main verb, but can be the matrix predicate as a whole:

- (10) *Mildred was thrown in the lion cage to keep her from talking.*

⁶ Notice that in sentences like (6a), it is the status of *Mary* as indirect object which allows it to control the subject gap. Prepositional objects which are not indirect objects cannot be controllers here, as shown in (a) below, while indirect objects which are not prepositional objects may still control the PC subject ((b) below).

a. *I_j got the bones; from Paul_k [e_j*_k to feed e_i to the dog]*
b. *I_j gave Mary_k this very dull book; [e_{*j}/_k to read e_i]*

Further, the RatC subject gap is optional:

- (11) *Elroy killed Oscar in order for Sylvia to escape.*

Finally, RatC are daughters of S, and not VP, and may therefore be preposed alone (12b) or otherwise isolated from the VP (12c):

- (12) a. *Helga carries a hat pin to protect herself.*
b. *To protect herself, Helga carries a hat pin.*
c. *What Helga does to protect herself is carry a hat pin.*

2.3 Infinitival Relative Clauses

Infinitival Relatives (IR) are superficially very similar to purpose clauses, especially in the patterning of their gaps. Like tensed relatives, they are daughters of NP; if the NP in question is in the VP, IR can be easily mistaken for PC:

- (13) a. IR: *I bought [a pan; [e to fry omelets in e_i]]*
b. PC: *I bought [a pan_i; [e to fry omelets in e_i]]*
- (14) a. IR: *Elroy really needs [a woman; [e_i to hold his hand]]*
b. PC: *Elroy really needs [a woman_i; [e_i to hold his hand]]*

IR, like PC, have one obligatory gap in either object (13a) or subject (14a) position, which is controlled, not by the matrix object (as in PC), but by the head of the NP containing the relative (just as in a tensed relative clause). If the obligatory gap is in object position, there may or not be a subject gap as well. This optional subject gap is controlled exactly like the optional gap in a PC.

An IR may be distinguished from a PC by making its containing NP the subject of the matrix sentence; PC may not occur in post-subject position. Another test is to make pro-nominal or definite the antecedent of the obligatory gap; IR may only have indefinite heads.

2.4 What the constructions mean

Three things are being communicated when one uses a purpose clause: an event of acquisition or use, an object (the thing which is being acquired or used), and the purpose to which the object will be put. That these elements form a deliberate complex and are not independent is made clear by attempting to omit either of the first two elements while retaining the syntactic form that gives the purpose clause its special character. In (15a), the object is not explicit in the matrix clause; in (15b), the matrix does not convey any sense of possession. Both are ungrammatical.

- (15) a. **I went to the bookstore to read on the plane.*
b. **Peter read a book for Helga to read on the plane.*

The infinitival relative, in contrast, has only two elements: an object and its purpose. Furthermore, there is no particular event that this purpose is specific to, i.e. no special relationship between the matrix clause in which the object appears and the purpose expressed by the relative.⁷ Consequently, the notion of purpose in an IR is narrower than in a PC, closer to the object's intrinsic function or unmarked use.

The rationale clause differs from both the PC and IR by not being object centered at all. Instead, a RatC adjunct expresses the goal which the matrix action was intended to bring about.

Note that as the various types of infinitive clauses become less deeply embedded, syntactically speaking, the scope of the expressed purpose becomes wider: from the standard function of an object, expressed within a noun phrase (IR); to the function some agent has imposed on an object, expressed in the verb phrase (PC); to the intended goal of the agent in performing the matrix activity, expressed in an S-level adjunct (RatC).

3. GENERATING THE CONSTRUCTIONS

To analyse a construction for generation, we must consider what it means, or, put another way, consider why a speaker would choose to use it, especially given the subtleties of meaning that differentiate it from similar constructions. The next consideration, and the subject of the present section, is how the construction should be situated within the generation process: what decisions, made at what point or points in the process, contribute to the selection and realization of the construction as part of an utterance? We begin with an overview of how decision making is organized in our model of generation. We then look at how a descriptive treatment in terms of thematic roles could be turned into an algorithm for generation, and show that it fails to take advantage of the information that is available at the early stages of generation. A treatment tailored to generation is markedly simpler: creating a PC from a motivated message is easier than describing the end product.

⁷ An NP containing an infinitival relative clause is characteristically descriptive rather than referential; however, this has more to do with the restrictive nature of the relative than with the content of the matrix.

3.1 Decision Making in Generation

In generation, unlike comprehension, the speaker's appreciation of his situation, his goals, and the information that he wants to communicate are self-evident, rather than needing to be discovered. The core problem in generation is making decisions: knowing what decisions must be made, what information bears on them, what the alternative choices are and how they are to be represented.

Carefully controlling the timing of when specific decisions are made offers the possibility of designing the generation process so as to achieve a very high level of efficiency. Forcing a decision too early before all of the information it requires is available may lead to guessing and later having to back up and undo that choice and any later ones that depended on it. Making a decision too late can mean missing opportunities to propagate information about the choice to other decisions that it should influence. Overall, the most pivotal and least constrained decisions should be made first, so that their consequences can be known soon enough to not hold up the others that are dependent on their choices.

In our model of generation, this criterion has led us to the view that decisions about the information an utterance is to convey will be made before decisions about syntactic form or serial order. These early decisions typically include choices of wording and influence all aspects of a text's form. The output of such decisions is expressed in an explicit representational level we call the "message level" (McDonald & Vaughan, 1987). Decisions reflecting the surface ordering of the arguments are made in the mapping to the next level of representation, the surface structure. As this structure is traversed, decisions about the particular realization of the arguments are made, morphological specialization is done, and the text is output.

3.2 Attempting to Adapt a Descriptive Analysis

In conventional transformational-generative analyses, the rules governing the occurrence of gaps in the constructions we are studying are characterized from a purely descriptive perspective. They do not try to determine which argument should be gapped, but rather where gaps may occur and what the antecedent of each gap will be. Directly adapting such an analysis to the generation task would involve complete specification of the surface structure followed by a multi-step matching algorithm to realize the gap(s).

Of descriptive analyses, those couched in terms of thematic roles seem best suited for the generation of PC, since they allow a single description of the antecedent of the obligatory gap. A possible algorithm for locating gaps in PC would be as follows (assuming that arguments of the matrix verb are still accessible from within the adjunct and are annotated with their thematic roles):

1. Gap the first argument in the PC which is an occurrence of the matrix Theme.
2.
 - a. If the PC subject matches the matrix Goal, gap it; or
 - b. if there is no matrix Goal and the PC subject matches the matrix Source or Location, gap it; or
 - c. if there is no matrix Source or Location either, and the PC subject is given as "unspecified", gap it.

While for our purposes such an algorithm is an improvement over a structural description, it is still unnecessarily complicated. For instance, there is no need to search the matrix clause for its theme since when generating we already know trivially which argument to obligatorily gap--the one that the purpose clause was chosen to express the purpose of.

3.3 Choosing the construction

Since, as we have discussed, there are semantic differences among PC, IR and RatC, the choice among them is more than just stylistically motivated syntactic variation. This means that they will be distinguished at the message level, since that is where an utterance's information content is determined. We have also argued that the PC and its matrix clause form a conceptual unit centering around the object whose use is in question. If that integrity is not to be left to chance, that conceptual unit must be chosen as a piece, making the PC an atomic resource that the English language provides, like adjectives or the copular clause.

At the message level then, we have a three part relation embodied in a "realization specification" (see example in Section Four), which stipulates that the statement of possession or access to an object and the statement of the purpose of that object are to be realized as main clause and PC respectively, with the occurrence of the object in the PC realized as a trace. The obligatory gap is thus inserted at the message level, and persists into surface structure, where realization of the two clauses as active, passive, etc. can take place without a subsequent costly calculation

of which structural position should be realized as a gap. (We will discuss the optional gap below). Since the tense of the adjunct is left unspecified in the realization specification, it will surface as an infinitive.

Delaying the realization of the two clauses until the linguistic context governing that realization has been established provides versatility. For example, the whole construction could be a complement to another verb, as in (16a), or to another infinitival adjunct, such as the rationale clause shown in (16b):

- (16) a. *I wanted to buy a book to read on the plane.*
- b. *I went to the bookstore to buy a book to read on the plane.*

One potential problem with this analysis is that the lack of prior constraint leaves open the possibility of generating rather awkward constructions, such as the following:

- (17) *A book was bought by me to read on the plane.*

It is our intuition, however, that the awkwardness of this sentence comes from a lack of motivation for the passive rather than any problem with the construction as a whole. Without a motivated source, this construction would never be generated; consequently we need not address how to block it. We can use this sort of argument to great advantage when working in a generation framework, which is one of the reasons why it provides a better model of how language is actually produced than the usual linguistic strategy of free generation with surface level filters.

The obligatory gap in the PC can (and should) be handled at the message level because (1) at that point all the information it requires is available, (2) no further information bearing on the identification of the argument to be gapped will become available later during realization (i.e. there is nothing gained by waiting), and (3) the means for carrying out the gapping operation are at hand (see next section). The optional subject gap is a different matter. This gap is licensed only if its antecedent is explicitly mentioned in the main clause, a fact that is not known at the message level. (More to the point, having known the information when the message was being assembled was unlikely to have changed the decisions that were made; consequently there is no utility to making it explicit there.)

Since the information needed to consider gapping the PC's subject is not available until the matrix clause has been realized, the gapping operation must be done at the level of surface structure rather than the message level. By relying on the fact that only well-formed, motivated messages are ever going to be

constructed, a surface-level rule for the operation can be compactly stated: "gap if the subject is mentioned in the matrix or is arbitrary (and non-emphatic)." 8

The single gap of a rationale clause is handled very much like the optional gap of the purpose clause. The planner is responsible for the overall relationship between an action and an intended result of that action. When the message is converted to a surface structure, it is realized as a main and a subordinate clause; the main clause, as the head of the bundle, is built first, and the RatC is then attached either before or after it. During traversal of the tree, the RatC subject will be gapped if it matches the main clause subject or the main clause as a whole. Once again, the information needed to determine whether to gap is not available until late in the process.

4. EXAMPLE

In this section we describe the particulars of our implementation of purpose clauses in the natural language generation system, Mumble. As we discussed in the previous section, this construction originates from a three part relation between an event, an object, and its purpose. At the message level, the interface to Mumble, the schema shown below in Figure One takes these three arguments and builds a realization specification for a purpose clause:

```
(define-specification-schema
  object-centered-event-&-purpose
  (object event object-purpose)
  (let ((matrix (instantiate-specification
    event)))
    (adjunct (instantiate-specification
      object-purpose)))
  (add-further-specification matrix
    :specification adjunct
    :attachment-function 'purpose-of)
  (locate-argument-&-force-to-a-trace
    object :containing-rspec adjunct) ))
```

FIGURE ONE

8 "Emphatic" refers to both marked stress, as in (a), or an unusual situation, as in (b), where the possessor is not the intended user:

- a. *I bought that dinosaur for me to play with (so keep your hands off it!)*
- b. *I bought David a dinosaur for me to play with (when I go over to his house).*

In a generation model, which assumes the generator is working in the service of some coherent underlying program, the information of when something is emphatic, or marked, is always known and can be made available to the linguistic processes, and would be necessary in any event in order to generate speech.

Figure Two shows the pretty printing of the realization specification created by this schema in order to generate the following text:

```
"Floyd bought Helga a book to read on the plane."
(event-bundle
  :head (:realization-fn buy
    1      :arguments (#<Floyd> #<Helga>
                        #<book> ))
  2:accessories ( tense-modal past )
  3:further-specifications
    (:specification
      (event-bundle
        :head (:realization-fn read
          6      :arguments
            (#<Helga> 4
              (:trace #<book>)) )
        :further-specifications
          (#<on-location #<read ...>
            #<plane> > )
      5 :attachment-fn purpose-of )))
```

FIGURE TWO

In order to make the example clearer, we have used the short hand notation #< ... > to indicate an underlying object from which a specification will be planned, rather than writing out its specification in all its detail. In the context of an actual underlying program generating from internally modeled objects, these could be unplanned specifications of objects, with planning and realization interleaved. However, as this example presently runs in our "stand-alone" interface, all the details are spelled out in the realization specification.

The bundle representation allows the planner to group component parts of the utterance. The head of the bundle (#1) is a constraint expression specifying the matrix clause. Accessories (#2) contain linguistically marked information, such as tense and NP number. The further-specification field (#3) specifies the adjunct. Note that the argument for #<book> (#4) has already been constrained to be a trace. The attachment function (#5) indicates how the further specification is related to the head. In this instance the attachment function is the particular attachment point PURPOSE-OF (shown in Figure Three), which splices a new element, labeled FOR-INFINITIVE, into the surface structure as the last element of the VP.

```
(define-attachment-point purpose-of
  :splice
  :reference-labels (vp)
  :link (last)
  :new-slot (for-infinitive) )
```

FIGURE THREE

Every specification has a realization function and a list of arguments. In general, the realization function is a class of choices which defines the set of initial trees (Joshi, 1985) which can realize the specification. The choices are annotated with the grammatical and contextual characteristics which distinguish their use. For example READ (#6), through a curried realization class (shown in Figure Four), uses the class AGENT-VERB-THEME⁹

```
(define-curried-realization-class Read
  (agent theme)
  :class agent-verb-theme
  ((verb "read")))

(define-realization-class Agent-verb-theme
  (agent verb theme)
  ( ((basic-clause-svo agent verb theme)
     (clause) () )
    ((for-infinitive-svo agent verb theme)
     (for-infinitive) () )
    ((relative-clause-svo rel-pro(agent)
      trace(agent)
      verb
      theme) )
    (relative-clause)
    (arg-same-as-head(agent)) )
    ((relative-clause-svo rel-pro(theme)
      agent
      verb
      trace(theme)) )
    ...
  ))
```

FIGURE FOUR

The message is realized in stages. First, the head of the bundle (#1) is realized by making a choice in its class (similar to that for READ in Figure Four) and building the surface structure representation for that choice, shown below in Figure Five.

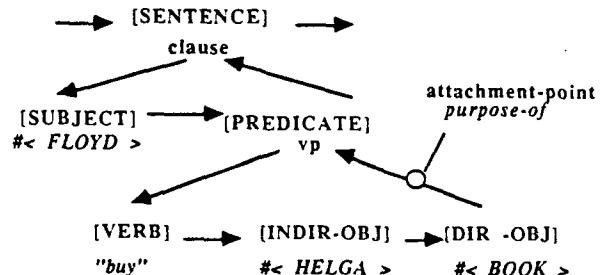


FIGURE FIVE

⁹ We use thematic roles as argument names in classes heuristically; we are not committing ourselves at this point to a thematic analysis of argument structure.

The accessories (#2) are then processed, which in this example sets the tense. Next the further specification is spliced in using the indicated attachment point. The surface structure is traversed, and embedded arguments are realized as they are reached. The result of these operations and the traversal up to the subject of the adjunct is shown in Figure Six. Note that the text of the main clause has been morphologically specialized and output as a side effect of the traversal, the surface structure for the argument NPs has been chosen and built, and the trace for the obligatory gap is already in place.

At this point the optional subject gap of the adjunct is considered. Since "Helga" is available as an explicit argument of the matrix clause, the subject is realized as a trace and the "for" is suppressed.

5. RELATED WORK IN GENERATION

Derr & McKeown (1984) directly address the generation of complex sentences; however, they restrict the criteria for combining propositions to focus and shared arguments. While it is fairly clear that they could extend their analysis to allow combinations based on relations between propositions that are expressible as explicit lexical connectives, it is unclear as to whether they could easily extend it to relations expressed structurally. They assume that the propositions are independently determined before possibilities for combinations are considered. While a special device could determine whether the particular relation licensing a PC was intended, they would lose the advantage we gain from letting the initial choice of object and construction be made

simultaneously. They would have to use an algorithm such as the one described in the thematic analysis above to determine the gapping pattern of the adjunct.

Davey (1974) and Kukich (1985) both simplify their approach to the problem by completely predetermining how propositions may be combined into complex sentences. Kukich uses predefined phrases and Davey a set of rules particular to the annotated move list of the tic-tac-toe game he is generating from. While these approaches provide an opportunity for choosing structures such as purpose clauses early and as one piece, they are seriously lacking in generality and flexibility. Both assume a limited domain where all of the possible propositions and their plausible combinations can be predetermined.

In the Knowledge Delivery System (KDS) Mann & Moore (1981) use a hillclimbing algorithm to determine which propositions should be combined into complex sentences. The algorithm assumes the information to be conveyed has been broken into kernel sized chunks and filtered to delete any repetitious or inferable information. This has the drawback that once the original information has been fragmented into kernels, the original relations between them have been lost. The aggregation rules must consequently use shared arguments and predefined templates to combine the kernels into sentence sized chunks. This causes the same problems as those described for Derr & McKeown: determining the gapping pattern in the adjunct clause and retaining generality.

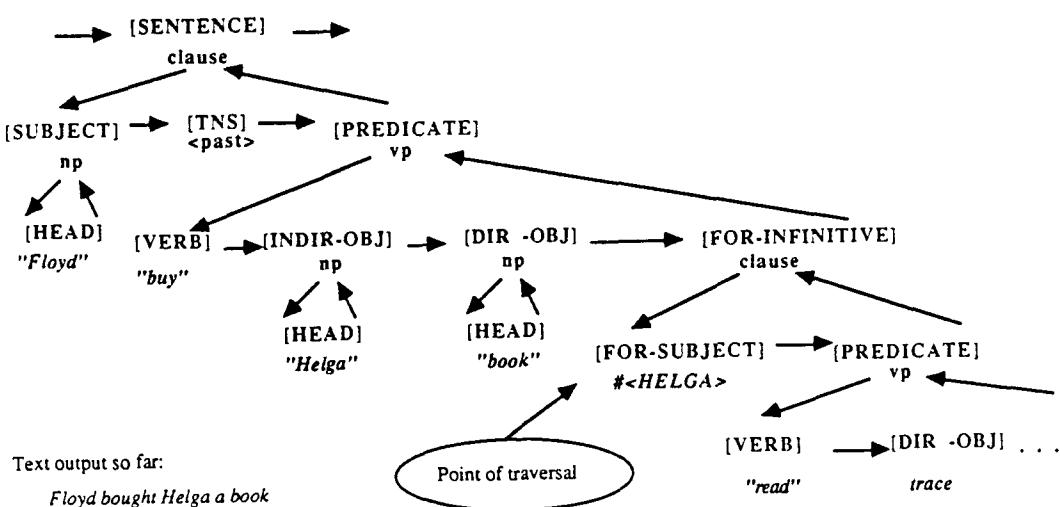


FIGURE SIX

6. CONCLUSION

In this paper we have shown the importance of carefully choosing the framework in which to couch one's analysis. For the generation of adjunct clauses, a computational approach which assumes a coherent underlying world model and text planner has clear advantages over a descriptive representation. We have also shown advantages of our model of generation: Our use of a message level distinct from and prior to the surface structure representation allows decisions to be made when germane information is most naturally available.

7. REFERENCES

- Bach, Emmon (1982), "Purpose Clauses and Control." In Jacobson & Pullum, eds., *The Nature of Syntactic Representation*, Reidel, Dordrecht, pp. 35-57.
- Chomsky, Noam (1980), "On Binding." *Linguistic Inquiry* 11.1, MIT Press, Cambridge.
- Davey, Anthony (1974), *Discourse Production*. Edinburgh University Press, Edinburgh, U.K. "Using Focus to Generate Complex and Simple Sentences." *Proceedings of Coling-84*, pp.319-326.
- Faraci, Robert A. (1974), *Aspects of the Grammar of Infinitives and For Phrases*. MIT Doctoral Dissertation (unpublished).
- Halliday, M.A.K. & Ruqaiya Hasan (1976), *Cohesion in English..* London: Longman Group Ltd.
- Huettnner, Alison K. (1987), *Adjunct Infinitives: An Exegesis*. PhD Dissertation, University of Massachusetts, Amherst, Massachusetts, forthcoming.
- Jespersen, Otto (1940), *A Modern English Grammar on Historical Principles*. G. Allen & Unwin, London.
- Jones, Charles (1985), *Syntax and Thematics of Infinitival Adjuncts*. PhD Dissertation, University of Massachusetts, Amherst, Massachusetts.
- Joshi, Aravind (1985), "Tree Adjoining Grammars: How much context-sensitivity is required to provide reasonable structural descriptions?" In Dowty, Karttunen, & Zwicky (eds.), *Natural Language Parsing*, Cambridge University Press, Cambridge.
- Kukich, Karen (1985), "Explanation Structures in XSEL." *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, pp. 228-237.
- Ladusaw, William & David Dowty (1985), "Towards a Formal Semantic Account of Thematic Roles." Unpublished manuscript.
- Mann, William & James Moore (1981), "Computer Generation of Multi-paragraph English Text." *American Journal of Computational Linguistics*, Vol. 7, No.1, Jan.-Mar., pp.17-29.
- McDonald David D. (1984), "Description Directed Control: Its implications for natural language generation." In Cercone (ed.), *Computational Linguistics*, Plenum Press, pp. 403-424; reprinted in Grosz, Sparck Jones, & Webber, *Readings in Natural Language Processing*, Morgan Kaufman Publishers, California, 1986.
- McDonald David D. & Marie M. Vaughan (1987), "Arguments for a Message Level in Natural Language Generation." Submitted to IJCAI-87.
- Nishigauchi, Taisuke (1984), "Control and the Thematic Domain." *Language* 60, Linguistic Society of America, Waverly Press, Baltimore.
- Rappaport, Malka & Beth Levin (1986), "What to do with Theta Roles." Lexicon Project Working Papers #11, MIT Center for Cognitive Science, Cambridge.
- Ritchie, Graeme (1984), "A Rational Reconstruction of the Proteus Sentence Planner." *Proceedings of Coling-84*, pp. 327-329.
- Williams, Edwin (1980), "Predication." *Linguistic Inquiry* 11.1, MIT Press, Cambridge.