TAG's as a Grammatical Formalism for Generation

David D. McDonald and James D. Pustejovsky

Department of Computer and Information Science
University of Massachusetts at Amherst

1. Abstract

Tree Adjoining Grammars, or "TAG's", (Joshi, Levy & Takahashi 1975; Joshi 1983; Kroch & Joshi 1985) were developed as an alternative to the standard syntactic formalisms that are used in theoretical analyses of language. They are attractive because they may provide just the aspects of context sensitive expressive power that actually appear in human languages while otherwise remaining context free.

This paper describes how we have applied the theory of Tree Adjoining Grammars to natural language generation. We have been attracted to TAG's because their central operation—the extension of an "initial" phrase structure tree through the inclusion, at very specifically constrained locations, of one or more "auxiliary" trees—corresponds directly to certain central operations of our own, performance-oriented theory.

We begin by briefly describing TAG's as a formalism for phrase structure in a competence theory, and summarize the points in the theory of TAG's that are germaine to our own theory. We then consider generally the position of a grammar within the generation process, introducing our use of TAG's through a contrast with how others have used systemic grammars. This takes us to the core results of our paper: using examples from our research with well-written texts from newspapers, we walk through our TAG inspired treatments of raising and wh-movement, and show the correspondence of the TAG "adjunction" operation and our "attachment" process.

In the final section we discuss extensions to the theory, motivated by the way we use the operation corresponding to TAG's' adjunction in performance. This suggests that the competence theory of TAG's can be profitably projected to structures at the morphological level as well as the present syntactic level.

2. Tree Adjunction Grammars

The theoretical apparatus of a TAG consists of a primitively defined set of "elementary" phrase structure trees, a "linking" relation that can be used to define dependency relations between two nodes within an elementary tree, and an "adjunction" operation that combines trees under specifiable constraints. The elementary trees are divided into two sets: initial and auxiliary. *Initial trees* have only terminals at their leaves. *Auxiliary trees* are distinguished by having one non-terminal among their leaves; the category of this node must be the same as the category of the root. All elemental trees are "minimal" in the sense that they do not recurse on any non-terminal.

A node N1 in an elementary tree may be linked (co-indexed) to a second node N2 in the same tree provided N1 c-commands N2. Linking is used to indicate grammatically defined dependencies between nodes such as subcategorization relationships or filler-gap dependencies. Links are preserved (though "stretched out") when their tree is extended through adjunction; this is the mechanism TAG's use to represent unbounded dependencies.

Sentence derivations start with an initial tree, and continue via the adjunction of an arbitrary number of auxiliary trees. To adjoin an auxiliary tree A with root category X to a initial (or derived) tree T, we first select some node of category X within T to be the point at which the adjunction is to occur. Then (1) the subtree of T dominated by that instance of X (call it X') is removed from T, (2) the auxiliary tree A is knit into T at the position where X' had been located, and (3) the subtree dominated by X' is knit into A to replace the second occurence of the category X at T's frontier. The two trees have now been merged by "splicing" A into T, displacing the subtree of T at the point of the adjunction to the frontier of A.

For example we could take the initial tree:

$[_{S'}$ Who$_i$ does $[_S$ John like e_i]]

(the subscript "i" indicates that the "who" and the trace "e" are linked) and adjoin to it the auxiliary tree:

$[_S$ Bill believes S]

to produce the derived tree:

$[_{S'}$ Who$_i$ does $[_S$ Bill believe $[_S$ John likes e_i]]]

Adjunction may be "constrained". The grammar writer may specify which specific trees may be adjoined to a given node in an elementary tree; if no specification is given the default is that there is no constraint and that any auxiliary tree may be adjoined to the node.

2.1 Key features of the theory of TAG's

A TAG specifies surface structure. There is no notion of derivation from deep structure in the theory of TAG's—the primitive trees are not transformed or otherwise changed once they are introduced into a text, only combined with other primitive trees. As Kroch and Joshi point out, this means that a TAG is incomplete as an account of the structure of a natural language, e.g. a TAG grammar will contain both an active and a passive form of the same verbal subcategorization pattern, without an theory-mediated description of the very close relationship between them.

To our minds this is by no means a deficit. The procedural machinery that generative grammars have traditionally carried with them to characterize relations like that of active to passive has only gotten in the way of employing those characterizations in processing models of generation. This is because a generation model, like any theory of performance, has a procedural structure of its own and cannot coexist with an incompatible one, at least not while still operating efficiently or while retaining a simple mapping from its actual machine to the virtual machine that its authors put forward as their account of psycholinguistic data.

Our own generator uses surface structure as its only explicitly represented linguistic level. Thus grammatical formalisms that dwell on the rules governing surface form are more useful to us than those that hide those rules in a deep to surface transformational process.

A TAG involves the manipulation of very small elementary structures. This is because of the stipulation that elementary trees may not include recursive nodes. It implies that the sentences one sees in everyday usage, e.g. newpaper texts, are the result of many sucessive adjunctions. This melds nicely with a move that we have made in recent years to view the conceptual representation from which generation proceeds as consisting of a heap of very small, redundantly related information units that have been deliberately selected by a text planning process from the total state of the knowledge base at the time of utterance; each such unit will correspond in the final text to a head lexical item plus selected thematic arguments—a linguistic entity that is easily projected onto the elementary trees of a TAG.

TAG theory includes only one operation, adjunction, and otherwise makes no changes to the elementary trees that go into a text. This comports well with the indelibility stipulation in our model of generation, since selected text fragments can be used directly as specified by the grammar without the need for any later transformation. The composition options delimited by the constraints on adjunction given with a TAG define a space of alternative text forms which can correspond directly in generation to alternative conceptual relations among information units, alternatives in rhetorical intent, and alternatives in prose style.

3. Adapting TAG's to Generation

The mapping from TAG's as a formalism for competence theories of language to our formalism for generation is strikingly direct. As we described in Section 5 their adjunction operation corresponds to our attachment process; their constraints on adjunction correspond to our attachment points; their surface structure trees correspond to our surface structure trees.[1] We further hypothesize that two quite strong correspondence claims can be made, though considerably more experimentation and theorizing will have to be done with both formalisms before these claims can be confirmed.

1. The primitive information units in realization specifications can be realized exclusively as one or another elementary tree as defined by a suitable TAG, i.e. linguistic criteria can be used in determining the proper modularity of the conceptual structure.[2]
2. Conversely, for <u>any</u> textual relationship which our generator would derive by the attachment of multiple information units into a single package, there is a corresponding rule of adjunction. Since we use attachment in the realization of nominal compounds like "*oil tanker*", this has the force of extending the domain of TAG analyses into morphology. (See section 7).

4. The Place of Grammar in a Theory of Generation

To understand why we are looking at TAG's rather than some other formalism, one must first understand the role of grammar within our processing model. The following is a brief summary of the model; a more complete description can be found in McDonald & Pustejovsky [1985b].

---

[1] Our model of generation does not employ the simple trees of labeled nodes that appear in most theoretical linguistic analyses. Our surface structure incorporates the semantic properties of trees, but it also includes reifications of constituent positions like 'subject" or 'sentence" and is better characterized overall as an 'executable sequence of labeled positions". We discuss this further in section 5.1.

[2] If this hypothesis is sucessful, it has very consequential implications for the 'size" of the information units that the text planner constructing the realization specification can use, e.g. they would not be realized as texts that include recursive nodes. We will discuss this and other implications in a later paper.

We have always had two complementary goals in our research: on the one hand our generation program has had to be of practical utility to the knowedge based expert systems that use it as part of a natural language interface. This means that architecturally our generator has always been designed to produce text from conceptual specifications, "plans", developed by another program and consequently has had to be sensitive to the limitations and varying approaches of the present state of the art in conceptual representation.

At the same time, we want the architecture of the virtual machine that we abstract out of our program to be effective as a source of psycholinguistic hypotheses about the actual generation process that humans use; it should, for example, provide the basis for predictive accounts of human speech error behavior and apparent planning limitations. To achieve this, we have restricted ourselves to a highly constrained set of representations and operations, and have adopted strong and suggestive stipulations on our design such as high locality, information encapsulation, online quasi-realtime runtime performance, and indelibility.[3] This restricts us as programmers, but disciplines us as theorists.

We see the process of generation as involving three temporally intermingled activities: (1) determining what goals the utterance is to achieve, (2) planning what information content and rhetorical force will best meet those goals given the context, and (3) realizing the specified information and rhetorical intent as a grammatical text. Our *linguistic component* (henceforth LC), the Zetalisp program MUMBLE, handles the third of these activities, taking a "realization specification"[4] as input, and producing a stream of morphologically specialized words[5] as output.

As described in [McDonald 1984], LC is a "description-directed" process: it uses the structure of the realization specification it is given, plus the syntactic surface structure of the text in progress (which it extends incrementally as the specification is realized) to directly control its actions, interpreting them as though they were sequential computer programs. This technique imposes strong demands on the descriptive formalism used for representing surface structure. For example, nodes and category labels now designate actions the generator is to take (e.g. imposing scoping relations or constraining embedded decisions) and dictate the inclusion of function words and morphological specializations.

3 "Indelibility" in a computation requires that no action of a process (making decisions, constructing representations, changing state, etc.) can be transparently undone once it has been performed. Many nonbacktracking, nonparallel program designs have this property; it is our term for what Marcus [1980] referred to as the property of being "strictly deterministic".

4 A realization specification can informally be taken to correspond to what many researchers, particularly psychologists, think of as the "message level" representation of a text.

5 Which is to say that it presently produces written rather than spoken texts. We expect to work with speech output shortly, however, and the need to support the representational basis of an intonational contour is beginning to influence our designs for constituency patterns in surface structure.

4.1 Unbundling Systemic Grammars

Of the established linguistic formalisms, systemic grammar [Halliday 1976] has always been the most important to AI researchers on generation. Two of the most important generation systems that have been developed, PROTEUS [Davey 1974] and NIGEL [Mann & Matthiessen 1983], use systemic grammar, and others, including ourselves, have been strongly influenced by it. The reasons for this enthusiasm are central to the special concerns of generation. Systemic grammars employ a functional vocabulary: they emphasize the uses to which language can be put—how languages achieve their speakers' goals—rather than its formal structure. Since the generation process begins with goals, unlike the comprehension process which begins with structure, this orientation makes systemic grammars more immediately useful than, for example, transformational generative grammars or even procedurally oriented AI formalisms for language such as ATN's.

The generation researcher's primary question is why use one construction rather than another—active instead of passive, "the" instead of "a". The principle device of a systemic grammar, the "choice system", supports this question by highlighting how the constructions of the language are grouped into sets of alternatives. Choice systems provide an anchoring point for the rules of a theory of language use since it is natural to associate the various semantic, discourse, or rhetorical criteria that bear on the selection of a given construction or feature with the choice system to which the construction belongs, thus providing the basis of a decision-procedure for selecting from its listed alternatives; the NIGEL system does precisely this in its "chooser" procedures.

In our formalism we make use of the same information as a systemic grammar captures, however we have choosen to bundle it quite differently. The underlying reason for this is that our concern for psycholinguistic modeling and efficient processing takes precedence in our design decisions about how the facts of language and language use should be represented in a generator. It is thus instructive to look at the different kinds of linguistic information that a network of choice systems carry. In our system we distribute these to separate computational devices.

- Dependencies among structural features: A generator must respect the constraints that dependencies impose and appreciate the impact they have on its realization options: for example that some subordinate clauses can not express tense or modality while main clauses are required to; or that a pronominal direct object forces particle movement while a lexical objects leaves it optional.

- o Usage criteria. The decision procedures associated with each choice system are not a part of the grammar per se, although they are naturally associated with it and organized by it. Also most systemic grammars include very abstract features such as "generic reference" or "completed action", which cross-correlate the language's surface features, and thus are more controllers of why a construct is used rather than constructs themselves.
- o Coordinated structural alternatives. A sentence may be either active or passive, either a question or a statement. By grouping these alternatives into systems and using these systems exclusively when constructing a text, one is guaranteed not to combine inconsistent structural features.
- o Efficient ordering of choices. The network that connects choice systems provides a natural path between decisions, which if followed strictly guarentees that a choice will not be made unless it is required, and that it will not be made before any of the choices that it is itself dependent upon, insuring that it can be made indelibly.
- o Typology of surface structure. Almost by accident (since its specification is distributed throughout all of the systems implicitly), the grammar determines the pattern of dominance and constituency relationships of the text. While not a principle of the theory, the trees of clauses, NPs, etc. in systemic grammars tend to be shallow and broad.

We believe, but have not yet established, that equivalence transformations can be defined that would take a systemic grammar as a specification to construct the alternative devices that we use in our generator (or augment devices that derive from other sources, e.g. a TAG) by decomposing the information in the systemic grammar along the lines just listed and redistributing it.

5. Example Analyses

One of the task domains we are currently developing involves newspaper reports of current events. We are "reverse engineering" leading paragraphs from actual newspaper articles to produce narrow but complex conceptual representation, and then designing realization specifications—plans—that will lead our LC to reconstruct the original text or motivated variations on it. We have adopted this domain because the news reporting task, with its requirement of communicating what is new and significant in an event as well as the event itself, appears to impose exceptionally rich constraints on the selection of what conceptual information to report and on what syntactic constructions to use in reporting it (see discussion in Clippinger & McDonald [1983]. We expect to find out how much complexity a realization specification requires in order to motivate such carefully composed texts; this will later guide us in designing a text planner with sufficient capabilities to construct such specifications on its own.

Our examples are drawn from the text fragment below (Associated Press, 12/23/84); the realization specification we use to reproduce the text follows.

"LONDON - Two oil tankers, the Norwegian-owned Thorshavet and a Liberian-registered vessel, were reported to have been hit by missiles Friday in the Gulf.

The Thorshavet was ablaze and under tow to Bahrain, officials in Oslo said. Lloyds reported that two crewmen were injured on the Liberian ship."

```
(the-day's-events-in-the-Gulf-tanker-war
   :events-require-certification-as-to-source
   (main-event  #<same-event-type_varying-patient
                        #<hit-by-missiles  Thorshavet>
                        #<hit-by-missiles  Liberian> >
                :unusual   #<number-of-ships-hit  2>
                :identify-the-ships   )
   (particulars  #<damage-report  Thorshavet  Oslo-officials>
                 #<damage-report  Liberian  Lloyds>   ))
```

Figure 1

This realization specification represents the structured object which gives the toplevel plan for this utterance. Symbols preceded by colons indicate particular features of the utterance. The two expressions in parentheses are the content items of the specification and are restricted to appear in the utterance in that order. The first symbol in each expression is a label indicating the function of that item within the plan; embedded items appearing in angle brackets are information units from the current-events knowledge base.

Obviously this plan must be considerably refined before it could serve as a proximal source for the text; that is why we point out that it is a "toplevel" plan. It is a specification for the general outline of the utterance which must be fleshed out by recursive planning once its realization has begun and the LC can supply a linguistic context to further constrain the choices for the units and the rhetorical features.

For present purposes, the key fact to appreciate about this realization specification is how different it is in form from the surface structure. One cannot produce the cited text simply by traversing and "reading out" the elements of the specification as though one were doing direct production. Structural rearrangements are required, and these must be done under the control of constraints which can only be stated in linguistic vocabulary with terms like "subject" or "raising".

The first unit in the specification, #<same-event-type_>, is a relation over two other units. It indicates that a commonality between the two has been noticed and deemed significant in the underlying representation of the event. The present LC always realizes such relations by merging the realizations of the two units. If nothing else occurred, this would give us the text *"Two oil tankers were hit by missiles"*.

As it happens, however, a pending rhetorical constraint from the realization specification, events-require-certification-as-to-source will force the addition of yet another information unit,[6] the reporting event by the news service that announced the aledged event (e.g. a press release from Iraq, Reuters, etc.). In this case the "content" of the reporting event is the two damage-reports which have already been planned for inclusion in the utterance as part of the "particulars" part of the specification. Let us look closely at how that reportiing event unit is folded into surface structure.

When not itself the focus of attention, a reporting event is typically realized as "so-and-so said X", that is, the content of the report is more important than the report itself; whatever significance the report or its source has as news will be indicated subtlly through which of the alternative realizations below is selected for it.[7]

Desired characteristic	**Resulting text**
de-emphasize report	*Two tankers were hit, Gulf shipping sources said.*
source is given elsewhere	*Two tankers were reported hit.*
emphasize report	*Iraq reported it hit two tankers.*

Figure 2 Possibilities for expressing report(source, info) in newpaper prose

In our LC, these alternative "choices" are grouped together into a "realization class" as shown in Figure 3. Our realization classes have their historic origins in the choice systems of systemic grammar, though they are very different in almost every concrete detail. The most important difference of interest theoretically is that while systemic choice systems select among single alternative features (e.g. passive, gerundive), realization classes select among entire surface structure fragments at a time (which might be seen as prespecified realizations of bundles of features). That is, our approach to generation calls for us to organize our decision procedures so as to select the values for a number of linguistic features simultaneously in one choice where a systemic grammar would make the selection incrementally.[8]

```
(define-realization-class   believe-verbs
  : parameters   (agent proposition verb)
  : choices
     (( (AGENT-VERBs-that-PROP agent verb prop)
        clause  focus(agent)  emphasize(self) )
        ; e.g. "Lloyds reports Iraq hit two tankers."
        ; encompasses variations with and without that, and
        ; also tenseless complements like "John believes him
        ; to be a fool."
      ( (raise-VERB-into-PROP  (passivize verb) prop)
        clause  focus((agent prop))  mentioned-elsewhere(agent) )
        ; "Two tankers were reported to have been hit"
      ( (It-VERB-PROP   verb prop)
        clause  inferable(agent) )
        ; e.g. "It is reported that 2 tankers were hit."
      ( (left-dislocated-PROP   agent verb prop)
        clause  de-emphasize(self) )
        ; "Two tankers were hit, Gulf sources said."
     ))
```

Figure 3 Realization class assigned to report(Iraq,hit(...)

Returning to our example, we are now faced now with the need to incorporate a unit denoting the report of the Iraqi attacks into the utterance to act as a certification of the #<hit-by-missiles> events. This will be done using the realization class believe-verbs; the class is applicable to any information unit of the form **report(source, info)** (and others). It determines the realization of such units <u>both</u> when they appear in issolation and, as in the present case, when they are to augment an utterance corresponding to one of their arguments.

From this realization class the choice raise-VERB-into-PROP will be selected since (1) the fact that two ships were hit is most significant, meaning that the focus will be on the information and not the source (n.b. when the class executes the source Iraq will be bound to its parameter and the information about the missile hits to the proposition parameter); (2) there is no rhetorical motivation for us to occupy space in the first sentence with the sources of the report since they have already been planned to follow. These conditions are sensed by attached procedures associated with the characteristics that annotate the choice (i.e. focus and mentioned-elsewhere).

6 We will not discuss the mechanism by which features in the specification influence realization. Realization specifications of the complexity of this example are still very new in our research and we are unsure whether the process is better organized at the conceptual level directing a composition process within the planning component (during one of the recursive invocations) or within the LC mediating a selection between anticipated alternatives. At this point our design experiments are inconclusive.

7 These sentences are artificial; actual ones would be considerably longer. Interestingly, certain other syntactically permissable versions such as "*It was reported that*" do not occur in any of the texts we have examined. Perhaps the "lead NP" position is too important to waste on a pronoun.

8 This technique of using choice systems to control the active selection of utterance features is employed by the most well-known applications of systemic grammars to generation (i.e. the work of Davey [1974] and Mann and Matthiessen [1983]). However very recent work with systemic grammars at Edinburgh by Patten [1985] departs from this technique. Patten uses a semantic-level planning component to directly select groups of features at the rightward, "output", side of a systemic network, and then works backwards through the network to determine what other, not sematically specified features must be added to the text for it to be grammatical; control is thus outside the grammar proper, with grammar rules relegated to constraint specification only. We are intrigued by this technique and look forward to its further development.

Since the PROP is already in place in the surface structure tree, the LC will be interpreting raise-VERB-into-PROP as a specification of how it may fold the auxiliary tree for *reported* into the tree for *Two oil tankers were hit by missiles Friday in the Gulf*. This corresponds to the TAG analysis in Figure 4 [Kroch & Joshi 1985].

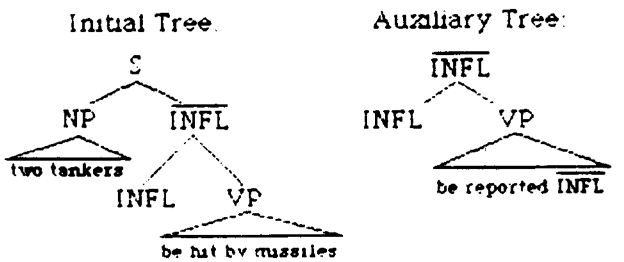

Figure 4 Initial and auxilliary trees for Raising-to-subject

The initial tree for *Two oil tankers were hit by missiles*, I_1, may be extended at its INFL′ node as indicated by the constraint given in parenthesis by that node. Figure 5 shows the tree after the auxiliary tree A_2, named by that constraint has been adjoined. Notice that the original INFL′ of Figure 4 is now in the complement position of report, giving us the sentence *Two oil tankers were reported hit by missiles*.

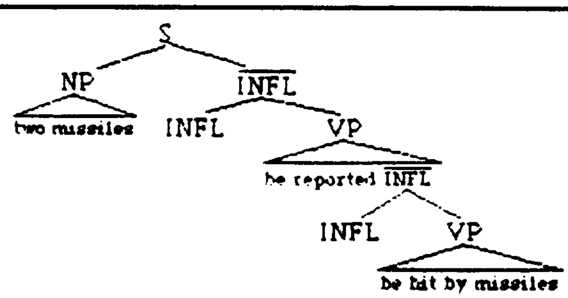

Figure 5 After embedding *report*

5.1 Path Notation

As readers of any of our earlier papers are aware, we do not employ a conventional tree notation in our LC. A generation model places its own kinds of demands on the representation of surface structure, and these lead to principled departures from the conventions adopted by theoretical linguists. Figure 6 shows the surface structure as our LC would actually represent it just before the moment when the adjunction is made.

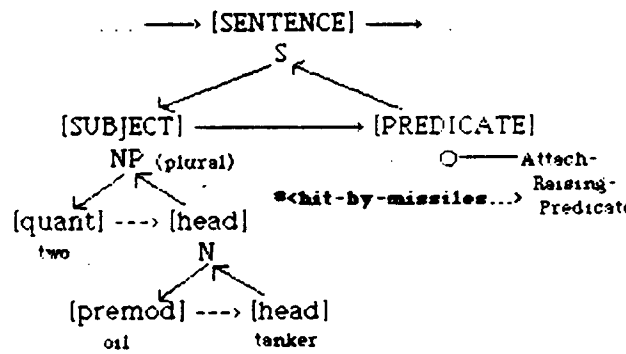

Figure 6 Surface structure in path notation

We call this representation *path notation* because it defines the path that our LC. Formally the structure is not a tree but a uni-directional linked list whose formation rules obey the axioms of a tree (e.g. any path "down" through a given node must eventually pass back "up" through that same node). The path consists of a stream of entities representing phrasal nodes, constituent positions (indicated by square brackets), instances of information units (in boldface), instances of words, and activated attachment points (the labeled circle under the predicate; see next section). The various symbols in the figure (e.g. **sentence**, **predicate**, etc.) have attached procedures that are activated as the point of speech moves along the path, a process we call "phrase structure execution". Phrase structure execution is the means by which grammatical constraints are imposed on embedded decisions and function words and grammatical morphemes are produced (For discussion see McDonald [1984]).

Once one has begun to think of surface structure as a traversal path, it is a short step to imagining being able to cut the path and "splice in" additional position sequences.[9] This splicing operation inherits a natural set of constraints on the kinds of distortions that it can perform, since, by the indelibility stipulation, existing position sequences can not be destroyed or rethreaded. It is our impression that these constraints will turn out to be formally the same as those of a TAG, but we have not yet carried out the detailed analyses to confirm this.

9 The possibility of cutting the surface structure and inserting new sequences that change the linguistic context of positions already in place has been in our theory of generation since 1978, when we used it to implement raising verbs whose rhetorical force was the same as "hedging" adverbs like *possibly*. Our present, much more extensive use of this device as the core of a distinct attachment process dates from the summer of 1984.

10 Constraints of this sort are an inovation introduced in Kroch & Joshi [1985]. Previous versions of TAG theory allowed "context sensitive" constraint specifications that in fact were never exploited. The present constraints are more attractive formally since they must be stated locally to a single tree.

5.2 Attachment Points

The TAG formalism allows a grammar writer to define "constraints" by annotating the nodes of elementary trees with lists indicating what auxiliary trees may be adjoined to them (including "any" or "none").[10] In a similar manner the "choices" in our realization classes—which by our hypothesis can be taken to always correspond to TAG elementary trees—include specifications of the *attachment points* at which new information units can be incorporated into the surface structure path they define. Rather than being constraints on an otherwise freely applying operation, as in a TAG, attachment points are actual objects interposed in the path notation of the surface structure. A list of the attachment points active at any moment is maintained by the attachment process and consulted whenever an information unit needs to be added. Most units could be attached at any of several points, with the decision being made on the basis of what would be most consistent with the desired prose style (cf. McDonald and Pustejovsky [1985a]). When one of the points is selected it is instantiated, usually splicing in new surface structure in the process, and the new unit added at a designated position within the new structure. Figure 7 shows our present definition of the attachment point that ultimately leads to the addition of "was reported".

```
(define-attachment-point   attach-raising-predicate
  reference-points
     ((present-predicate (slot-contents 'predicate phrase)))
  required-characteristics-of-unit's-realization
     ( raising-verb-with-complement(present-predicate) )
  position-of-attachment-point
     ( (actual-slot 'predicate phrase)
       :attach-under )
     new-phrase-structure
       (submerge-existing-contents-into-new-structure
         (vp-infinitive-complement) ; specification of new phrase
          verb    ; where the unit being attached goes
          infinitive-complement)    ; where the existing contents go
  effect-on-other-pending-attachment-points
     none
  choices-that-introduce-it
     :choices-passing-test (includes-slot 'predicate))
```

Figure 7 The attachment-point used by *was reported*

This attachment point goes with any choice (elementary tree) that includes a constituent position labeled **predicate**. It is placed in the position path immediately after (or "under") that position (see Figure 6), where it is available to any new unit that passes the indicated requirements.

When this attachment is selected, it builds a new VP node that has the old VP as one of its constituents, then splices this new node into the path in its place as shown in Figure 7.

The unit being attached, e.g. the report of the attack on the two oil tankers, is made the verb of the new VP. Later, once the phrase structure execution process has walked into the new VP and reached that verb position, the unit's realization class (**belief-verbs**) will be consulted and a choice selected that is consistent with the grammatical constraints of being a verb (i.e. a conventional variant on the **raise-VERB-into-PROP** choice), giving us "*was reported*".

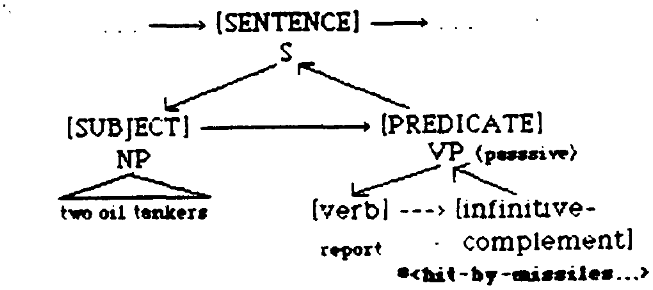

Figure 8 The path after attachment

From this discussion one can see that our treatment of attachment uses two structures, an attachment point and a choice, where a TAG would only use one structure, an auxiliary tree. This is a consequence of the fact that we are working with a performance model of generation that must show explicitly how conceptual information units are rendered into texts as part of a psycholinguistically plausible process, while a TAG is a formalism for competence theories that only need to specify the syntactic structure of the grammatical strings of a language. This is a significant difference, but not one that should stand in our way in comparing what the two theories have to offer each other. Consequently in the rest of this paper we will omit the details of the path notation and attachment point definitions to facilitate the comparison of theoretical issues.

6. Generating questions using a TAG version of wh-movement

Earlier we illustrated the TAG concept of "linking" by showing how one would start with an initial tree consisting of the *innermost* clause of a question plus the fronted wh-phrase and then build outward by successively adjoining the desired auxiliary phrases to the S node that intervenes between the wh-phrase and the clause. Wh-questions are thus built from the bottom up, as in fact is <u>any</u> sentence involving verbs taking sentential complements.

This analysis has the desirable property of allowing one to state the dependencies between the *Wh*-phrase and the gap as a local relation on a single elementary tree, eliminating the need to include any machinery for movement in the theory. All unbounded dependencies now derive from adjunctions (which, as far as the grammar is concerned, can be made without limit), rather than to the explicit migration of a constituent across clauses.

We also find this locality property to be desirable, and use an analogous procedure in our production of questions and other kinds of *Wh*questions and unbounded dependency constructions.

This "bottom-up" design has consequences for how the realization specifications for these constructions must be organized. In particular, the logician's usual representation of sentential complement verbs as higher operators is not tenable in that role. For example we cannot have the source of, say, *How many ships did Reuters report that Iraq had said it attacked?* be the expression:

Lambda(quantity-of-ships). report

(Reuters,say(Iraq,attack(Iraq,quantity-of-ships)))

Such an expression defines a natural *sequence of exposure* when used as realization specification, namely that one realize the **Lambda** operator first, the **report** operator second, the **say** third, and so on. A local TAG analysis of *Wh*-movement requires us to have the **Lambda** and the expression containing its matrix trace, **attack**, be present in a single "layer" of the specification, otherwise we would be forced to violate one of the strong principles of our theory of generation, namely that the characteristics in a realization class may "see" only the immediate arguments of the unit being realized; they may not look "inside" those arguments to subsequent levels of conceptual structure.

This principle has served us well, and we are disinclined to give it up without a very compelling reason. We elected instead to give up the internal representation of sentential complement verb texts as single expressions. This move was easy for us to make since such expressions are awkward to manipulate in the "East Coast" style frame knowledge bases that we use in our own reasoning programs, and we have preferred a representational style with redundant, smaller sized conceptual units for quite some time.

The representation we use instead amounts to breaking up the logical expression into individual units and allowing them to include references to each other.

$$U_1 = \text{lambda(quantity-of-ships) . attack(Iraq,quantity-of-ships)}$$

$$U_2 = \text{say}(\text{Iraq}, U_1)$$

$$U_3 = \text{report}(\text{Reuters}, U_2)$$

Given such a network as the realization specification, the LC must have some principle by which to judge where to start: which unit should form the basis of the surface structure to which the others are then attached? A natural principle to adopt is to begin with the "basis" unit, i.e. the one that does not mention any other units in its definition. We are considering adopting the policy that such units should be allowed only realizations as initial trees while units whose definition involves "pointing to" (naming) other units should be allowed only realizations as auxiliary trees. We have not, however, worked through all of the ramifications such a policy might have on other parts of our generation model; without yet knowing whether it would improve or degrade the other parts of our theory, we are reluctant to assert it as one of our hypotheses relating our generation model to TAG's.

Given that three part source, the realization of the question is fairly straightforward (See Figure 9). The **Lambda** expression is assigned a realization class for clausal *Wh* constructions, whereupon the extracted argument **quantity-of-ships** is placed in COMP, and the body of the expression is placed in the HEAD position. At the same time, the two instances of **quantity-of-ships** are specially marked. The one in COMP is assigned to the realization class for *Wh* phrases appropriate to quantity (e.g. it will have the choice *how many X* and possibly related choices such as <quantity> *of which* and other variants appropriate to relative clauses or other positions where *Wh* constructions can be used). Simultaneously the instance of **quantity-of-ships** in the argument position of the head frame **attack** is assigned to the realization class for *Wh*-trace. These two specializations are the equivalent, in our model, of the TAG linking relation.

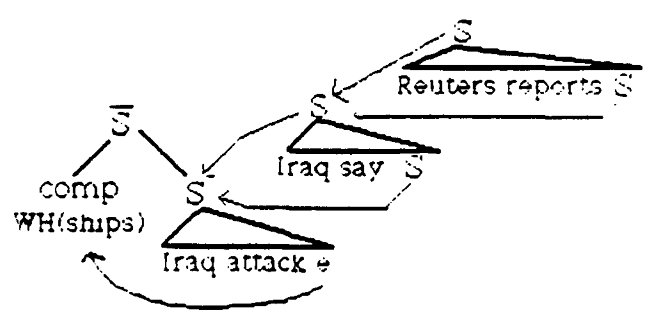

Figure 9 Question formation with sentential complement verbs

The two pending units, U_2 and U_3, are then attached to this matrix, submerging first the **attack** unit and then U_2 into complement positions.

7. Extensions to the Theory of TAG

Context-free grammars are able to express the word formation processes that seem to exist for natural languages (cf. Williams [1981], Selkirk [1982]). A TAG analysis of such a grammar seems like a natural application to the current version of the theory (cf. Pustejovsky (in preparation)). To illustrate our point, consider compounding rules in English. We can say that for a context-free grammar for word formation, G_w, there is a TAG, T_w, that is equivalent to G_w (cf. Figures 10 and 11). Consider a fragment of G_w below.[11]

11 Whether the word formation component should in fact have the power of a TAG or CFG is an open question. Langendoen [1981] discusses the possibility that a finite state grammar might be sufficient for the generative capacity of natural language word formation components.

N -> N | A | V | P N
A -> N | A | P A
V -> P V

Figure 10 CFG Fragment for Word Formation

The corresponding G_w fragment would be:

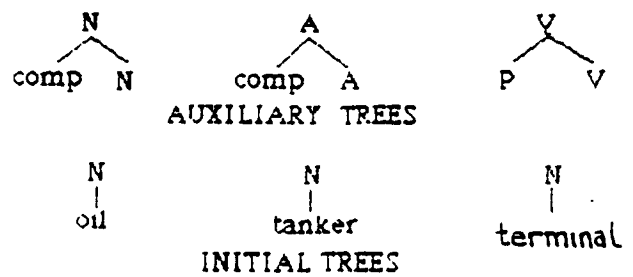

Figure 11 TAG Fragment for Word Formation

Now consider the compound , *"oil tanker terminal*, taken from the newspaper reporting domain, and its derivation in TAG theory, shown in Figure 12.

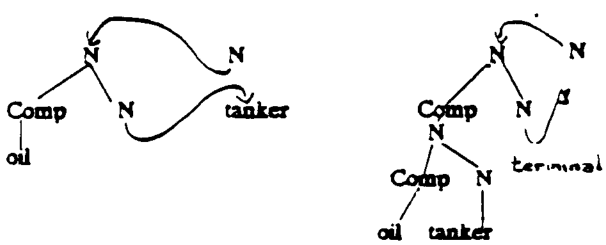

Figure 12 TAG Derivation of *oil tanker terminal*

Let us compare this derivation to the process used by the LC. The underlying information units from which this compound is derived in our system are shown below. The planner has decided that the units below need to be communicated in order to adequately express the concept. The top-level unit in this bundle is #<terminal>.

U_1 = #<terminal>
U_2 = #<docks-at U_1 U_3>
U_3 = #<tanker>
U_4 = #<carries U_3 U_5>
U_5 = #<oil>

The first unit to be positioned in the surface structure is U_1, and appears as the head of an NP. There is an attachment point on this position, however, which allows for the possibility of expressing U_2 prenominally. One of the choices associated with this unit is a compound structure—expressed in terms of an auxiliary tree. A snapshot at this point in the derivation shows the following structure.

$[_N\ [_{Comp}\ U_2\ ]\ \ U_1\ ]$

The next unit opened up in this structure is U_3, which also allows for attachment prenominally. Thus an auxiliary tree corresponding to U_4 is introduced, giving us the structure below:

$[_N[_{Comp}\ [_{Comp}\ U_4\ ]\ U_3]]\ U_1]$

The selectional constraints imposed by the structural positioning of information unit U_4 allows only a compounding choice. Had there been no word-level compound realization option, we would have worked our way into a corner without expressing the relation between #<oil> and #<tanker>. Because of this it may be better to view units such as U_4 as being associated directly with a lexical compounded form, i.e. *oil tanker*. This partial solution, however, would not speak to the problem of active word formation in the language. Furthermore, it would be interesting to compare the strategic decisions made by a generation system with those planning mistakes made by humans when speaking. This is an aspect of generation that merits much further research.

8. Acknowledgements

This research has been supterminaled in part by contract N0014-85-K-0017 from the Defense Advanced Research Projects Agency. We would like to thank Marie Vaughan for help in the preparation of this text.

9. References

Clippinger, & McDonald (1983) "Why Good Writing is Easier to Understand", Proc. IJCAI-83, pp. 730-732.

Davey (1974) **Discourse Production**, Ph.D. Dissertation, Edinburgh University; published in 1979 by Edinburgh University Press.

Halliday (1976) **System and Function in Language**, Oxford University Press.

Joshi (1983) "How Much Context-Sensitivity is Required to Provide Reasonable Structural Descriptions: Tree Adjoining Grammars", preprint to appear in Dowty, Karttunen, & Zwicky (eds.) **Natural Language Processing: Psycholinguistic, Computational, and Theoretical Perspectives**, Cambridge University Press.

Kroch, T. and A. Joshi (1985) "The Linguistic Relevance of Tree Adjoining Grammar", University of Pennsylvania, Dept. of Computer and Information Science.

Langendoen, D.T. (1981) "The Generative Capacity of Word-Formation Components", **Linguistic Inquiry**, Volume 12.2

Mann & Matthiessen (1983) *Nigel: A Systemic Grammar for Text Generation*, in Freedle (ed.) **Systemic Perspectives on Discourse**, Ablex.

Marcus (1980) *A Theory of Syntactic Recognition for Natural Language*, MIT Press.

McDonald (1984) "Description Directed Control: Its Implications for Natural Language Generation", in Cercone (ed.) **Computational Linguistics**, Pergamon Press.

McDonald & Pustejovsky (1985a) "SAMSON: a computational theory of prose style in generation", Proceedings of the 1985 meeting of the European Association for Computational Linguistics.

__________ (1985b) "Description-Directed Natural Language Generation", Proceedings of IJCAI-85, W.Kaufmann Inc., Los Altos CA.

Patten T. (1985) "A Problem Solving Approach to Generating Text from Systemic Grammars", Proceedings of the 1985 meeting of the European Association for Computational Linguistics.

Pustejovsky, J. (In Preparation) "Word Formation in Tree Adjoining Grammars"

Selkirk (1982) **The Syntax of Words**, MIT Press.

Williams (1981) "Argument Structure and Morphology" *The Linguistic Review*, 1, 81-114.