Description-Directed Natural Language Generation

David D. McDonald and James D. Pustejovsky

Department of Computer and Information Science
University of Massachusetts at Amherst

1. Abstract

We report here on a significant new set of capabilities that we have incorporated into our language generation system MUMBLE. Their impact will be to greatly simplify the work of any text planner that uses MUMBLE as ita linguistics component since MUMBLE can now take on many of the planner's text organization and decision-making problems with markedly less hand-tailoring of algorithms in either component. Briefly these new capabilies are the following:

(a) ATTACHMENT. A new processing stage within MUMBLE that allows us to readily implement the conventions that go into defining a text's intended prose style, e.g. whether the text should have complex sentences or simple ones, compounds or embedding*, reduced or full relative clauses, etc. Stylistic conventions are given as independently stated rules that can be changed according to the situation.

(b) REALIZATION CLASSES are a mechanism for organizing both the transformational and lexical choices for linguistically realizing a conceptual object. The mechanism highlights the intentional criteria which control selection decisions. These criteria effectively constitute an "inteiiingua" between planner and linguistic component, describing the rhetorical uses to which a text choice can be put while allowing its lingustic details to be encapsulated.

The first part of our paper (sections 2 and 3) describes our general approach to generation; the rest illustrates the new capabilities through examples from the UMass COUNSELOR Project. This project is a large new effort to develop a natural language discourse system based on the HYPO system [Rissland & Ashley 1964], which acts as a legal advisor suggesting relevant dimensions and case references for arguing hypothetical legal cases in trade-secret law. At various relevant points we briefly contrast our work with that of Appelt, Danlois, Gabriel, Jacobs, Mann and Mattheissen, and McKeown and Derr.

2. Major Components of the Generation Process

As nearly everyone who has worked on generation from an AI perspective will agree, it is the character of the decision-making involved, and not any *a priori* division by linguistic level, that dictates how the process divides into components. Decision-making varies in at least the three dimensions listed below, with the clustering patterns of groups of decisions along these dimensions deterining what different components there should be.

o What information does the decision draw on: properties of lexical items? Conceptual attributes? Details of planned rhetorical structures? Details of surface structures?

o What is the decision dependent on: what other generation decisions, if made differently, would force a change in this decision? If the generation process is to be indelible (i.e. never retracting its decisions) then this dependency structure will have to be respected.

o How should the decision's conclusions be represented: does a conclusion dictate linguistic actions or just constrain other decisions? Can it be acted on immediately or must it be scheduled for later?

On the basis of such considerations, we have determined that generation as a whole involves three different kinds of activities: two that are dominated by conceptual (typically domain-specific) criteria and decisions, and one that is dominated by linguistic criteria and has a correspondingly wider applicability. These three activities are:

1. determining what goals to (attempt to) accomplish with the utterance;
2. deciding what information the utterance should convey and what rhetorical force it should have in order to satisfy those goals;
3. realizing those conceptual "specifications" as a grammatical text that is cohesive with the discourse that precedes it.

We see these activities as intermingled and ongoing throughout the duration of a text's generation. Goals are only partially formulated when one starts speaking, and may even emerge opportunistically as the linguistic structure of the text is incrementally planned and produced. We presume that in people these activities are carried out in parallel, though in our discourse system for the COUNSELOR project they are treated as strictly gated co-routines.

It is the last of these activities, the linguistically most demanding, that is our concern in this paper This aspect of our research has two goals, one pragmatic, the other scientific First, for researcher* who need capable natural language interfaces, we have developed a versatile *linguistic component* ("1C"), implemented as the Zetalisp program MUMBLE, that can be interfaced to domain specific systems, where it handles all of the linguistic considerations that occur during generation The domain specific system supplies a specification of what it wants said, couched in its own internal representation, and is then realized by our LC as a grammatical text

Our second goal is to project from the computational architecture of our LC to hypotheses about the actual generation process in human beings. To this end we have disciplined ourselves in the design of the computer program to employ only devices with specific, very narrow capacities. This influences our choice of linguistic analyses, in that only a few of the analyses that one could imagine (and appear in the literature) are plausible in our generator, allowing us to develop linguistic and psycholinguistic hypotheses that are predictive and falsifiable. We will not discuss this aspect of our research in this paper, some of our earlier conclusions arc described in [McDonald 1984]

3. Our Approach to Generation

In this section we describe our approach in general terms and outline the separate components of our system; detailed examples will be given in the section following.

3.1 Goal -directed Generation

In our view, research on the production of text is most revealing when it considers how the text advances the goals of a speaker. A generation program should produce texts for an audience that is situated in a concrete discourse context working from a particular plan of information to be communicated with specific rhetorical goals Furthermore, we believe (and it is here that we part company with researchers such as Mann and Matthiessen [1983], whose aims we otherwise share) that the demands placed on an LC by the need to work efficiently from a plan have overriding implications for the LC s architecture This can require it to take a form very different from what is developed when working on the form/function relationship in isolation Generation programs like Mann and Matthiessen s NIGEL that have so far been used only to produced well-formed texts in Isolation from computationally represented situations or speakers' goals may tell us a great deal about the structure of grammars and linguistic *competence,* but it does not follow that they are therefore *models* of the generation process

In accordance with this methodology, our LC has always been developed in the context of one or more *underlying programs* which have been responsible for dictating the communicative goals. Each program has included a *text planning system,* of varying (sometimes trivial) sophistication (For example the program GENARO developed by Jeff Conklin [1983] and the programs listed in [McDonald 1983].) The text planners react to their associated programs' communicative goals and construct a conceptual level "plan"—a non-linguistic specification of what it wants said—which is given to the LC as its input.

Although we arc presently working with text planners that we are designing ourselves, we expect that the place we have picked for the division between planning what to say and determining how to say it is a natural one; consequently our Linguistic Component should be able to be used to advantage with text planners of very different internal design than the ones we happen to have used.

32 Multiple levels in the linguistic component

An ongoing trend in our research has been to introduce more and more levels within the generation processes between originating the goals and ultimately producing the text At the present time we employ three sucessive levels of processing within the LC: attachment, realization, and phrase structure execution. We will briefly define each of these levels later in this section and will expound on their functional organization and give illustrations of the structures they use in the second half of the paper

At each processing level there is a shift in the vocabulary of the rules, the representations that are used, and the character of the reasoning that is brought to bear Each level is narrowly restricted in the kind of information that it can draw on and the textual horizon it is able to work within Each is a specialized activity within the "virtual machine" that we have developed for generation, and works according to its own computational principles and therefore with the high degree of efficiency that is possible through specialization of the mechanism to the task (provided, of course, that the "boundaries" between the levels are in fact well placed). Decisions at each level are simpler than if the levels were folded together, since less linguistic detail is determined at a time.

In this regard we disagree with Appelt [1980, 1985] about the benefits of maintaining a single computational paradigm throughout the generation process (in his case axiomatic planning): his point that all of the generation processes involve planning is very well taken, yet this need not imply a homogeneous design. Indeed, one of the implications of homogeneity is that all information is equally available at any point in the process, which we do not believe is the case in generation by people or is of any advantage in machine systems. With our multi-level design we can specify quite exactly when a given kind of information becomes available, whether it is available "early" in some more abstract form (ie. before the representation that would normally carry that information has been instantiated), and when it later becomes inaccessible.

It is crucial to understand that while we speak loosely of these different reasoning engines and representations as constituting "levels**, they are in fact not strictly ordered in the process as a whole. All of them are active simultamously, including the planner that supplies their input; they operate as closely coordinated co-routines, precisely synchronized to the "point of speech", passing information (in the form of progressively more refined specifications and eventually syntactic and lexical stuctures) from one to the other along a well defined path or "pipeline".

Setting goals and planning content. Generation starts when the underlying program sets the goals of the utterance. In our own work this is done by the decisions of a "discourse controller" designed after Woolf [1984]. The goals direct the actions of a conceptual or "information" level planner that makes decisions about what information to actually communicate (versus leave for the audience to infer), how to structure it as a text, and what rhetorical or discourse effects to achieve. The planner also occasionally selects key lexical items (anticipating and preempting later choices) when this allows the combination of basic information units into a more useful "package" (e.g. merging an object's temporal BEGINNING and ENDING fields by using the word "between", or using a term such as "get revenge" rather than simply stating the underlying events of a story [Cook, Lehnert, & McDonald 1984].) Figure 3 shows an example of a plan of the sort we are presently using.

To the LC, the output of the text planner appears as a sequence of kernal information units (i.e. objects from the internal conceptual representation of the underlying program, typically pointers to frames or relations extracted from them, see Figures 2 and 3). Each unit is typically accompanied by a rhetorical annotation, put there by the text planner, which describes some perspective or emphasis that the unit's textual realization is to reflect.

SURFACE STRUCTURE While there are four processes in our model of generation (planning, attachment, realization, and phrase structure execution), there are only two reference structures: the plan, and surface structure. This second representational level is the only linguistic level we support; its representational vocabulary is abstract and for the most part syntactic and phrasal. It corresponds to the level in generative grammars that is the input to the phonological component and incorporates many of the devices of modern linguistic theory such as "traces" and X-bar categories.

The surface structure is the core of our LC's design It defines the action sequence (program) carried out by Phrase Structure Execution, absorbs the output (in sucessive chunks) of Realization, and is the target of Attachment. The fact that a linguistically motivated description of an utterance in progress can take on such a controlling, coordinating role in the generation process is to us too striking to be a fortuitous accident of our skills as computer programmers, rather it reflects something crucial about the nature of the generation process operating in people. Space does not permit a proper technical description of our surface structure or of PSE; interested readers should see [McDonald 1984].

The ATTACHMENT PROCESS The first thing that must happen to the units in the plan is for them to be assigned ("attached") to positions within the surface structure. This extends the surface structure by instantiating one of the "attachment points" that have been assigned to it according to the grammatical structure it currently has (e.g. next-sentence, additional-Adjective embedded-discourse-unit etc.).

The process does not attach all of the units in a plan at once; instead attachment is interleaved with Phrase Structure Execution so that most earlier units will have been realized and their text spoken before the last one is positioned. To judge where to attach a unit, the process considers first the syntactic form of the various text choices that could be made in realizing the unit, filtering out any that are incompatible with the available attachment points (e.g. one cannot position an adjective to serve the syntactic function of a new sentence). It then employs a set of rules to order the remaining points according to the designated prose style.

The REALIZATION PROCESS Once a unit has been positioned within the surface structure, the next thing that happens to it is the selection of a text that will adequately "realize" its information content. This requires sensitivity to its functional role within the knowledge base and the plan (eg topic versus instrument), to the desired perspective, and to the grammatical constraints composed by the surface structure at that position.

The association of objects (or object types) from the underlying system's knowledge base with the texts that could be used to realize them is made by assigning them to predefined "realization classes". Realization classes are highly parameterized and annotated lists of "choices" where each choice defines a possible syntactic structure and wording. Each choice is annotated by a set of "characteristics" which summarize the linguistic nature of the choice and the functional uses to which it may be put The characteristics can also be viewed as predicates against the current state of the surface structure and plan and thus provide the backbone of the decision procedure that makes the realization choices Linguistically, the pattern of choices in a realization class most strongly resembles the "transformation families" of Harris [1951, 1952]

Choices are defined in terms of a schematically described phrase structure plus a "mapping" that takes the parameters of the choice into positions in the phrase. This parameterization reflects the compositionality of the units in the plan: most units can be viewed as relations over other units; a typical realization is then the selection of a verb to realize the relation with the argument units mapped into the verb't thematic roles

PHRASE STRUCTURE EXECUTION The phrase structure defined by a choice consists of a labeled tree of constituents with English words and some units as its leaves. The depth-first traversal of this tree defines a path that will enumerate the words and embedded units (which will then be realized and replaced by a phrase) in their natural left to right order as a text

Using this path as its controlling representation, Phrase Structure Execution ("PSE") carries the whole LC forward: Words at the leaves of the surface structure are morphologically specialized and spoken as soon as they are reached When embedded units are reached they are passed to the Realization Process; the sub-tree selected and instantiated by that process is then passed back to PSE and incorporated at the position of the unit, replacing the unit and thereby defining an extension of the traversal path. When a potential attachment point is reached the Attachment Process is awakened to determine whether it wishes to use that point for positioning the next unit in the plan

4. Some Examples

Consider the text below, which our LC generates from the plan in Figure 3 (as fleshed out by later subplanning). This text was originally produced by a human lawyer as the initial description of the legal case on which he was seeking advice, our research goal has been to reproduce it (and the rest of the dialogue it was part of—work which is not yet complete) in a generalizable way as a precursor to having the full COUNSELOR system eventually plan other such texts on its own

" I represent a client named RCAVICTIM who wants to sue SWIPEINC and Leroy Soleil for misappropriating trade secrets in connection with software developed by my client. RCAVICTIM markets the software known as AUTOTELL, a program to automate some of a bank teller's functions, to the banking industry. In 1982, Leroy Soleil, one of RACVICTIM's personnel, left RCAVICTIM and began working for SWIPEINC on a competing product, TELLERMATIC, also an automated teller program. SWIPEINC has begun marketing TELLERMATIC in competition with AUTOTELL. "

Figure 1 Output text from the LC

4.1 Rhetorically-annotated Input from the Planner

At the moment, this paragraph is generated from the plan in Figure 2. This structure specifies the following. (1) The purpose of the utterance, i.e. to inform the audience of the intention to bring suit and the basic facts of the case. (2) Its information content—the frames from the knowledge base that will be the proximal sources for the wording of the text (given in the "#<...>" notation used to indicate flavor instances in ZetaLisp, the implementation language of our homebrew frame system). (3) Any special perspectives that are to govern the realization of those frames (indicated by the keyword ": perspective"). The specification of perspective is crucial because of the potentially enormous range of realizations most of the frames in this domain may have (eg a "neutral" perspective on this Legal-case should probably be realized as something like "RCAVICTIM is the plaintiff and SWIPEINC and Leroy Soleil are the defendants").

```
Inform
  : discourse-bridge
      #<Lawyer-Client-Relationship
              #<lawyer the-speaker>
              #<corporate-party RCAVICTIM>>
           : perspective establish-relation-of-speaker
  : head
      #<legal-case "RCAVICTIM vs. SWIPEINC">
           : perspective #<action-taken-by
                             #<corporate-party RCAVICTIM>>
  : motivation-behind-head
      #<claim misappropriation-of-trade-secret
                  #<knowledge-of #<product AUTOTELL>>>)
  : elaboration
      (event-history #<legal-case RCAVICTIM-us-SWIPEINC>)
           : perspective misapropriation-script
```

Figure 2 A Rhetorically-Annotated Plan

4.2 Incremental Planning

The text planning process has two problems: (1) to decide what information and rhetorical organization will best render the communicative intent of the underlying system, and (2) to bridge the gap between the way information is organized in that system's knowledge base and whar. is possible in the natural language being used (i.e. what are the words and grammatical constructions of the language). In our COUNSELOR project this gap is nontrivial, as illustrated by the frames in Figure 3. The dominating problem is the considerable difference in "packaging": the unit frames hold considerably more information than should be communicated at one time, making selection of perspective and other culling criteria quite important if what is said is to be relevant.

```
#<Legal-case
    name    "RCAVICTIM-vs-SWIPEINC"
    status   intended-but-not-brought-to-court
    party-list (#<corporate-party RCAVICTIM>
                #<corporate-party swipeinc>
                #<individual-party leroy-Soleil>)
    claim-or-defense-list
        (#<misappropriation-of-ts
                #<knowledge-of #<AUTOTELL>>) >

#<Corporate-party
    name `RCAVICTIM"
    case RCAVICTIM-vs-SWIPEINC
    case-role plaintiff
    employee-list (#<individual-party leroy-Soleil>)
    product-list (#<product autotell>)
    competitor-product-alist
        '((#<corporate-party rcavictim>
           #<product tellermatic>)) >
```

Figure 3 Example Frames from the Knowledge Base

Our planner, a program we are calling "CICERO", is presently quite limited. It is organized as a set of specialists with names like Describe-legal-case , Describe-a-party-to-a-case, Describe-corporate-party (a specialization of Describe-a-party-to-a-case) and so on. The procedure used now is a simple contextual discrimination between alternative, precompiled "scripts", which are then instantiated and passed to the LC in the form shown before in Figure 2. That plan is the standard one for unmarked descriptions of cases except for the addition of the first unit (indicating the relation of the speaker to the case), which was included because the utterance is the first one in the conversation and that relation cannot yet be deduced from context. We recognize that this use of scripts is quite limiting, and are in the midst of developing more versatile techniques.

One key part of CICERO's design which we expect to retain, however, is the fact that it operates incrementally. It would be neither psychologically plausible nor computationally economical to have a planner make all of the decisions about the information in an utterance (especially a long one) in "one pass**, and then completely relinquish control to the LC until it was time to plan the next utterance. Instead, planning should proceed incrementally, leaving the details of references or elaborations as conceptual "stubs" in the larger plan to be worked out in detail later once that plan has been partially realized and a linguistic characterization of that position of the stub becomes available. A recursive invocation of CICERO then uses that characterization, as well as the stub s position in the original plan, to aid it in deciding what information and perspectives to use. The resulting "subplan" is then passed back to the LC which continues where it left off.

So for example the plan in Figure 2 was constructed by Describe-legal-case. This specialist knows to mention the party (#<cooperate~parth Rcavictim>) that the lawyer is representing, but does not itself know how to describe that party, leaving that decision to be made later by Describe-corporate-party once PSE reaches the its position in the surface structure.

4.3 Attachment and Prose Style

The Attachment Process is a transducer from the stream of annotated units in the plan to assignments of active "attachment points" within the surface structure. These assignments modify the surface structure at the time they are made, in effect "splicing in" additional phrase structure to accomodate the new unit. What points are active at any moment is determined by the details of the surface structure already in place and the position of the PSE process. Deliberations over which attachment points to select are mediated by preferences which are characterized in terms of "stylistic rules"

At the beginning of an utterance, there is only one attachment point active, namely First-sentence. The first unit of the plan will be positioned here, and then almost immediately will be reached by PSE and passed to Realization. Realization will select a phrase for it (in this case a clause), and then knit the phrase into the surface structure in place of the unit.

```
[S #<speaker>
   [VP represent
        [NP #<corp.-party Rcavictim> ] ] ]
```

(The actual representation of surface structure is considerably more elaborate than this simplified presentation suggests.) Once this replacement occurs new attachment points are made active, as dictated by the form of the clause. These include New-sentence, various conjunctions such as Reduced-on-common-subject and Reduced-on-common-predicate, and especially various syntactic extensions to the final NP such as additional adjectives, postnominal phases, or quantifiers which would refine the characterization of Rca victim, as well as non-restrictive attachments relating some of its attributes or activities. What attachments are allowed is a function of the syntactic configuration of the surface structure ahead of the point of speech; the relevant criteria are discussed in [McDonald & Pustejovsky 1985b].

With the realization of the first unit, the attachment possibilities for the second can be considered. These are narrowed by ignoring all possibilities that demand a syntactic realization for a unit at their position that is not found in the second unit's realization class (which in the present case are essentially just variations on clauses involving the verb "sue"). At present this leaves only three points still in the running.

New-sentence: "I *represent a client named Rcavictim. They want to sue SwipeInc and Leroy Soleil."*

Simple-conjunction: *Y *represent a client named Rcavictim, and they want to sue SwipeInc and Leroy Soleil."*

Non-restjrictive-relauvcKAaining-off-final-NP: "I *represent a client named Rcavictim, who wants to sue SwipeInc and Leroy Soleil"*

Figure 4 Variations due to alternative attachment

We contend that the decision between these three is a matter only of the prose style one prefers: long sentences or short, simple syntax or complex. Lawyers have very definite preferences in their prose style which we can capture as a set of "ordering rules". These rules of prose style are used to order the alternatives, each rule accompanied by a predicate characterizing the linguistic contexts in which it applies. Our algorithms for this process are described in [McDonald & Pustejovsky 1985a].

```
(define-stylistic-rule   Always-prefer-extension-with-Relative-clause
  ordering-on-attachment-points
    (absolute-preference-for
          Non-restrictive-relative-chaining-off-final-NP  )
  applicability-condition
    (not  (is-more-than-one-sentence-before-point-of-speech
               (relative-text-position
                   (most-recent-use
                        'Non-restrictive-relative-chaining-off-final-NP)))))
```

Figure 5 A simple Stylistic Rule

After the non-restrictive-relative is selected, the surface structure will look approximately like this:

```
is #<speaker>
   [vp represent
        [ INP #<corp-party Rcavtetlm> ]
             [s #<toga>case ...> ] J ] ]
```

The standard technique for combining a sequence of conceptual units into a text has been "direct repfacemec!** (see discussion in [Mann et al 1982D, in which the sequential organization of the text is identical to that of the message because the message is used directly as a template. Our use of Attachment dramatically improves on this technique by relieving the message planner of any need to know how to organize a surface structure, letting it rely instead on explicitly stated stylistic criteria operating after the planning is completed

Den and McKeown [1984] also improve on direct replacement's one-proposition-for-one-sentencc forced style by permitting the combination of individual information units (of comparable complexity to our own) into compound sentences interspersed with rhetorical connectives. They were, however, limited to extending sentences only at their ends, while our Attachment Process can add units at <u>any</u> grammatically licit position ahead of the point of speech. Furthermore they do not yet express combination criteria as explicit, separable rules.

Dick Gabriel's program Yh [1984] produced polished written texts through the use of critics and repeated editing It maintained a very similar model to our own of how a text's structure can be elaborated, and produced texts of quite high fluency We differ from Gabriel in trying to achieve fluency in a single online pass in the manner of a person talking off the top of his head; this requires us to put much more of the responsibility for fluency in the pre-linguistic text planner, which is undoubtedly subject to limitations.

4.4 Realization Classes

We have said that language generation is a problem of how to organize decision making: first of what information to convey (done incrementally within the Text Planner), then of how to textually capture the relation of the units in a plan to each other (done by the Attachment Process), and finally of how to express—"realize"—the information in a conceptual unit, which is the task of Realization.

Realization is the selection of one of a predefined set of alternative *text* specifications ("choices") subject to grammatical constraints according to the position within the surface structure at which the realization occurs. The predefined choices are organized into "realization classes" such as shown in Figure 6. At the time this is written, these classes either organize alternative wordings, as in the concept-specific class or when wording has been determined, organize the structural (e.g. transformational) alternatives that the language permits, as in Transitive-Latinate-verbs

```
(define-realization-class   establish-relation-of-speaker-o-lawyer-to-client
  :parameters   (lawyer client)
  :choices
   (((represent-o-SVO lawyer client)
      ; I represent a client
     clause initial-mention(self) )
    ((relation-expressed-as-genitive ; my client
          lawyer (instance-of(client next-plan-step))))
     np instance-of(client (next-plan-step)) )) ))

(define-realization-class transitive-latinate-verbs-forming-nominalization
  : parameters   (verb subj obj)
  : choices
   (((ddefault-active-form verb subj obj)
      ; A misappropriates B
     clause )
    ((passive-form verb subj obj)
      ; B is misappropriated by A
     clause in-focus(obj) )
    ((gerundive-with-subject verb subj obj)
      ; A misappropriating B
     np)
    ((gerundive-passive-with-subject verb subj obj)
      ; B being misappropriated by A
     np in-focus(obj))
    ((nominalization (nominal verb) subj obj)
      ; A  misappropriation of B
     np expresses-result(verb) express(subj))
    ((definite-nominalization-with-subject nominal(verb) subj obj)
      ; the misappropriation of B by A
     np expresses-result(verb) )
    ((bare-nominalization-with-subject nominal(verb) subj obj)
      ; misappropriation of B by A
     np untensed-situation(self) ) ))
```

Figure 6 Two Realization Classes

The selection decision is based on the "characteristics" that accompany each choice (e.g. in-focus(arg), expresses-result(verb)). These symbols have attached procedures which test for prerequisites, and also may be tested for presence or absence by the grammatical constraints. Much of our ongoing research involves determining what reasonable characteristics are and what predicates they should be allowed to use; this will dictate the amount and character of the linguistic knowledge we allow our text planner to have, since all further information about a candiate text (for example the information used by PSE and morphology) is encapsulated in the choice.

It is appropriate at times to think of the choices in a realization class as items in a "phrasal lexicon" [Becker 1975], i.e. as productive, frozen turns of phrase that we use not because of their compositional, literal meaning, but because they are a conventional phrase rather like an idiom Our mechanism of associating conceptual units directly with choices (rather than mediating the concepts-text relationship with sets of abstract features, as for example done by Matthiessen [1983]) permits us to capture these conventional relationships easily, while noting those abstract relations that we do understand by labeling the choices with characteristics.

Linguistically, our choices are more versatile than the phrasal entries in use at Berkeley where phrasal lexicons are a common part of language processing programs, e.g. [Jacobs 1983]. Ours are transparent to the addition of other modifying or elaborating concepts within the phrase because of the thoroughness of the linguistic specification given with a choice and the use of the Attachment Process.

As one of the two loci of decision-making within the LC (the other being Attachment), we think of the process of Realization as simultaneously drawing on knowledge from syntactic, semantic, and pragmatic levels, which, as we stated at the very beginning of this paper is a natural property of the generation process. In this we agree with Danlois [1984], and see a familial resemblance between our realization dasses and the sets of cross-level alternatives she describes. We do, however, believe that a greater *a priori* ordering among decision dasses is possible than Danlois would permit. In particular it appears to us as more than just programmer's convenience is involved in our design decision to reduce the combinatoric complexity of realization dasses by having the localization decisions for phrasal heads made before alternative arrangements of thematic arguments are considered; we suspect that Danlois' motivation for her 'flatter*' design (i.e. more dasses of alternatives combined into a single decision) stems from the fact that she works from an underlying representation (Conceptual Dependency) that is less differentiated and more expression-based than ours.

5. Conclusions and Future Research

Our generation design now uses a significantly richer computational architecture than it has in the past; in particular, we have introduced the notions of an attachment process and realization dasses, both operating within a Description-Directed control structure coordinated by the Phrase Structure Execution Process. This has allowed us to produce a greater variety of texts with an intuitively more satisfying modularity in design (that is, the effects of stylistic rules, attachment points and realization dasses are limited and well-defined).

Our experience with this new architecture is still small, and our designs continue to evolve, especially as we now begin to work out a control structure for text planning that is not script based and to develop an executable statement of the compentence grammar that underlies the linguistic knowledge in the realization dasses and PSE. We do expect though, that the major outlines of what we have described will remain the same at least over the next several years A robust, well documented version of the system is under development for general release in the fall of 1986; preliminary versions are available now to people doing research on generation.

6. Acknowledgments

Funding for this research has been provided by DARPA under order number N00014-85-K-0017, by NSF grant 1ST 8104984 and Atari Corporation. We would like to thank Kevin Gallagher and Marie Vaughan for help in the preparation of the text.

7. References

Appelt D. (1980) "Problem Solving Applied to Language Generation"*, Proccdings of the 18th Annual Meeting of the Association for Computational Linguistics, June 19-22, 1980, University of Pennsylvania, pp.59-63.

Appelt D. (1981) "Planning Natural Language Utterances to Satisfy Multiple Goals" Ph.D. Thesis, Stanford University Available as SRI Technical Note 259.

Becker J. (1975) The Phrasal Lexicon" BBN Report No. 3081, Bolt Beranek and Newman, Inc., Cambridge, MA.

Conklin EJ (1983) "Data-Driven Indelible Planning of Discourse Generation Using Salience" PhD. Thesis, Department of Computer and Information Science, University of Massachusetts at Amherst; available as COINS Technical Report 83-13.

Cook M, Lehnert W., and McDonald D. (1984) "Conveying Implicit Context in Narrative Summaries" Proc. of COLING-84, Stanford University, pp5-7.

Danlos L. (1984) "Conceptual and Linguistic Decisions in Generation", Proceedings of COLIN G-84, Association for Computational Linguistics, pp501-504.

Derr M., A McKeown K., (1984)"Using Focus to Generate Complex and Simple Sentences" Proceedings of COLING-84, pp319-326.

Gabriel R., (184) Ph.D. thesis, Computer Science Department, Stanford University.

Harris Z (1951) Methods of Structural linguistics, Chicago, University of Chicago Press.

Harris Z. (1952) "Discourse Analysis: a sample text", Language 28, pp.474-494.

Jacobs P. (1983) "Generation in a Natural Language Interface," Proc. Eigth UCAI, pp.61(MS12.

Joshi A. (1983) "How much context-sensitivity is required to provide reasonable structural descriptions: Tree Adjoining Grammars" in Dowty, Karttunen, A Zwicky (eds) Natural Language Processings: Psychology, Computational, and Theoretical Perspectives, Cambridge University Press.

Mann W , Bates M., Grosz G., McDonald D., McKeown K , Swartout W., "Report of the Panel on Text Generation" Proceedings of the Workshop on Applied Computational Linguistics in Perspective, American Journal of Computational Linguistics, 8(2), pgs 62-70.

Mann W, A Matthiessen C, (1983) "A Systemic Grammar for Text Generation" IS1 Technical Report RR-83-105

Matthiessen C, (1983) "How to make grammatical decisions in text generation" ISI Technical Report RS-83-120.

McDonald D. (1983) "Natural Language Generation as a Computational Problem: an Intrduction" in Brady and Berwick, eds. Computational Problems in Discourse, MIT Press.

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

McDonald D., A Pustejovsky J. (1985a) "Samson: A computational Theory of Prose Style", Proceedings of the Conference of the European Association for Computational Linguistics, University of Geneva.

McDonald D., A Pustejovsky J. (1985b) ''TAGs as a Grammatical Formalism for Generation", Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics, University of Chicago.

Rissland E., Valcarce E., A Ashley K., "Explaining and Arguing with Examples" Proceedings of AAAI-84, pgs.288-294.

Woolf B. (1984) "Context Dependent Planning in a Machine Tutor" Ph.D. thesis. University of Massachusetts.