Computers for Composition: A Stage Model Approach to Helping

Lance A. Miller
IBM T. J. Watson Research Center, Yorktown Heights, New York

This paper is concerned with how computers can assist text composition. It is a review of what has been or is being done — particularly in the EPISTLE project at the Yorktown IBM Watson Research Center — and it is also a preview of what could be done in the future. The discussion is centered around a stage model of composition adapted from computer programming. The model's psychological validity, although credible, is secondary to the rather natural way it appears to organize possible computer function supporting composition. Some stages of this model clearly require powerful language technologies, and the stronger interest is clearly in these. Nevertheless, a number of functions are identified for other stages, which functions do not require extensive linguistic capability to implement or enhance. Throughout the paper there is a deliberate bias towards "practical" writing — writing whose fruits are of unquestioned value and whose authors receive monies undetermined by word counts or reviewers' praise. The notion of this type of writing — most frequent by far — is that the composition meets a requirement of business or social commerce, whether it is an inter-agency status report, product documentation, a step in the process of attempting to free a client, or a worded advertising fantasy designed to attract cash customers. The paper concludes with some consideration of specialized audiences for the various computer functions as well as the roles educators might play in promoting (or obstructing) their development.

A STAGE MODEL OF COMPOSITION

In this paper "composition" means a cohesive body of text expressed primarily in full grammatical constructions, with at least one primary communication objective, and usually embodied as a recognizable writing genre (such as a user manual, memorandum, or obituary). A composition is also the desired end product of the process of composition. While there are many important comments and observations on this activity (e.g., Flower 1981; Flower & Hayes 1980; Mathews & Stevenson 1976), there is certainly room for additional research, particularly as concerns detailed cognitive models. The idea of a "software-design" model applied to composing is one attractive possible extension.

The flavor of the software-design stage model is to identify a variety of stages for programming activities and to propose various tools and environ-

Visible Language, XX 2 (Spring 1986), pp. 188-218.
Author's address: IBM Federal Systems Division, 6600 Rockledge Drive, Bethesda, MD 20817.

ments to support each stage as well as the overall task (e.g., Yourdan & Constantine 1979; Orr 1977; Miller 1978). Although the stages are expressed as a linear sequence, a considerable amount of iteration among various stages is presumed throughout. Similarly, while the end-product appears to be better when all of the stages are passed through, in order, the model never purports to describe more than what some people do some of the time. A simplified form of the model is shown in Figure 1.[1] This simplified software development stage-model can be adapted almost directly to the process of composing by substituting some terms, as shown in Figure 2.

In stage 1 the problem to be solved by an eventual composition is considered and elaborated into a set of requirements which must pay at least some heed to the three primary rhetorical factors of audience (and speaker), topic, and communication purpose (e.g., Mills 1952). The additional rhetorical factor of the nature of construction of the actual message is not an aspect of this stage but of the next (or later).[2]

In stage 2 the initial decisions made for audience, topic, and pragmatic objectives are to be taken together to determine two emerging textual features: (1) the selection of the appropriate genre-format, and (2) the initial choice of style.[3] The first notion, genre-format, combines the selection of a particular writing genre (usually domain and intent specific — a "missive," a "persuasive argument") with a conventional schema for sequencing and emphasizing the parts thereof (a "business letter," a "court of appeals brief"). The second notion of style choice is taken here narrowly to mean the "strategy for communicating information," and it can operate both as a selector of genre-formats and as an organizer of material within a particular format.

1. Functional requirements
2. High-level design
3. Detailed design
4. Coding
5. Testing
6. Installation
7. Modification

Figure 1. Software production stage model

1. Communication requirements
2. High-level plan
3. Detailed plan
4. Scribing
5. Critiquing
6. Release
7. Adaption

Figure 2. Text composition stage model

The detailed structuring of a composition in stage 3 involves considerable fleshing out and elaborating facts, arguments, and themes — though still in pre-discourse form. Here more than elsewhere the items from unorganized lists ("points to make," "ground to cover," "essential information") get transferred into a detailed sequence, and the possibilities and necessities of coherence begin to become known.

If the process of composition is likened to a drama, then the theatrical climax would occur in the 4th stage, *scribing*. Up to this point, the planning and evaluation for composition could be quite similar to that for painting, writing music, or choreographing. Here, the words used as symbols for concepts of communication in extra-linguistic fashion now become actual text; the illocutionary intent is now instantiated as actual locution, capable now of that most dreaded of readers' acts — being quoted! No longer can a word or phrase merely stand for the intended sense, but *le mot juste* must be found; government, binding, and overall predication must be accomplished, with due consideration to all of the complexities of movement, raising, coordination, and agreement — and this just for a single sentence! How fearsome the choice then as inter-sentential organization must be achieved: here the strong marker of discourse structure ("First of all, . . ."), there the repeated theme by hint of lexical collocation (". . . the accident . . . the damage . . ."), and now the eschewing of anaphoric "he" for cataphoric ". . . these three:" Level after level must coordinate and cohere, topic to sentence, sentence to paragraph, paragraph to chapter — and all to each and every word. No wonder is it then that the slightest hesitation, the merest stickiness of fit, the briefest delay of word-finding all can bring such feverish (or labored) activity to an instant (or grudging) halt — to reconsider the last word, the format appropriateness, or even the whole stage 1 approach.

Such examinations of text, whether planned or forced, bring the process, however momentarily, into the 5th stage, *of critiquing*. Included are issues as mundane as the number of 'r's in "embarrass" or as complex as whether the main point is really being made. This stage is but a springboard, mostly, for revisiting previous stages. In particular, the visiting of the scribing stage after critiquing is called revision.[4]

The 6th stage, *release,* identifies the point at which the text leaves its maker and has an opportunity for a life of its own. Publishing of the text is one of the strongest forms of release (in several of its meanings!) although in general this occurs just whenever the text is truly "given away".

The last, the 7th stage, *adaptation,* refers to the modification of a document to reflect new requirements, new information to be included (updates), or error correction. The adaptation of previously released text almost always involves a new release of some sort — printing, copyright, or even published "errata." However, this stage is not an important aspect for the present focus.[5]

Stages to be discussed

The last two stages, while still important aspects of the composition process, go very far afield from the present topic of personal computer aids to writers. It's quite likely that the emerging Expert Systems technology could, some day, be

applied to these aspects, but that speculation is outside of this paper. The other stages are discussed roughly in terms of the extent to which computers presently support them. Thus, the fourth stage of scribing is discussed first, then stage 5 critiquing, and then stages 1-3.

COMPUTERS AND SCRIBING

Computers have had the most obvious impact in the scribing stage — the actual generation of text within some medium. Clearly, the most important role has been played by the word-processor, the first topic in this section. However, the role of formatters is also extremely important, and, finally, there are surely some number of yet unthought-of useful scribing aids.

Word-processors

Computer word-processors — and, lately, electronic typewriters with similar features — have revolutionized the process of business and technical writing. Perhaps the most glowing praise concerns not the actual transcription of words, but their editing — movement of text elements from one part to another, global changes, the location and modification of particular targets, and any of a host of other similar specialized functions (e.g., Sekuler 1985). These text-modification activities are still considered to be part of the scribing stage — particularly since they involve the computer editor; however, the thinking and evaluations that usually precede them are said to be of the following stage, critiquing.

The impact of computer word-processing on commercial scribing is all the more remarkable when it is considered that the computer software has no real idea of what the words mean or what the text is about. The functions operate on the notion of "token" — character strings delineated by blanks (or perhaps other special characters) — and the notion of "line record", a single line of the edited file as wide as permitted by the editor at that time. However, the editing capability provided by this primitive token analysis is quite adequate for most functions, and many users are not even aware that the editors do not share their view of word or sentence.[6]

It is not only the professional writers who have been influenced by this word-processor aspect of computers. There is also abundant evidence that word processing is welcomed eagerly by students in college and elsewhere, as the word-processing capability is introduced. There is a serious question for many, so we are told, whether they could ever write with pencil and paper again!

Computers used for the scribing phase are really inseparable from the question of the features available on the computer editor, and this has been an area of great interest to researchers, both from the standpoint of designing better editors (Miller & Thomas 1977; Embley & Nagy 1981; Roberts 1980) and from the point of view of editing as a psychological task (e.g., Card et al. 1976). The very large number of computer editors that are available — even from a single manufacturer — is indicative of the extent to which this area of the relation between computers and composition is being explored — and exploited.

Formatting

With pencil and paper and older typewriter technologies the act of scribing also — simultaneously — involved the act of deciding how to format the text on the page: whether to center, skip a line, indent, etc., for the grosser non-alphabetic manipulations; and, for the text itself, whether to capitalize, underline, change colors, or even change fonts. Thus, if a word was to be centered, capitalized, and underlined (to serve as a topic heading, for example), then these activities co-occurred with the actual generation of the characters of that word and the positioning of them on the paper.

Nowadays, with word-processors, you can defer the actual formatting until a later time by embedding special formatting codes in your text in special ways, and then sometime later sending your text file with these codes to a special formatting program to produce hard-copy or soft which has the text formatted according to your specifications; some of the formatting languages are SCRIBE, WATERLOO SCRIPT, and IBM SCRIPTVS. And, around the corner, are so-called WYSIWYG editors (pronounced "wiz-ee-wig" for "What You See Is What You Get") in which your typed formatting code is immediately executed, so that the text is formatted directly without a subsequent step. Nevertheless, since formatting was traditionally an integral part of the scribing process, and since formatting considerations still are very much a part of scribing, in terms of the requirement to embed formatting codes within the text being scribed, this activity is discussed here — and not, as it might well be, as part of the critiquing process.

Formatting technology has been around for a long time but is recently undergoing some important pervasive changes. Publishers are very near to agreeing to a standard for "marking up" documents in terms of classifying the contents of a particular body of text with certain "tags" preceding (and often ending) the text. This standard is being called Standard Generalized Markup Language or *SGML*. Thus, there are tags which identify the beginning of a document title, a footnote, a chapter heading, or an index. There are also tags which identify any arbitrary phrase, within any arbitrary document section, as, say, a definition, a citation, or simply something to be highlighted. For example, the reference to ". . . or *SGML* . . ." above (and here) was typed exactly as follows:

```
or :cit. SGML:ecit.
```

such that the acronym SGML was surrounded by two tags, each beginning with a":" and ending with a "." — the tag itself being "cit" for citation. Similarly, the first three numbered lines of Figure 1 were produced by the following set of formatting control tags embedded with the text:

```
:ol.
:li.Functional requirements
:li.High-level design
:li.Detailed design
```

The :ol. tag indicates that an ordered (numbered) list is to be started, and the :li. tag indicate "list-items" to be put one under another and numbered con-

secutively (see the IBM manual on the Generalized Markup Language, 1984).

These tags have two important advantages over first-generation formatting control words. First, they are nouns rather than verbs: they describe the nature of the text contained rather than specify what particular line-skipping, indenting, or other actions should take place. Thus, in the older IBM formatting language of SCRIPT, the control word .sk could be used to skip a line and start a new paragraph, without indentation; in the newer GML tag language, :p. is used instead. This symbol means that "the following is a new paragraph," but the actual interpretation is not specified by the author but by conventions observed at the local installation. In some situations, then, :p. could be interpreted as "skip a line and don't indent"; in another situation it could mean "skip and indent five spaces"; and in still a third — a publisher of children's stories, say — it could lead to that stylistic ballooning or enlarging of the first letter of the first new paragraph word. The author is therefore freed from having to think about ultimate formatting arrangements (see Misek-Falkoff 1983, for detailed discussion of these points).

A second advantage of the use of "noun" tags is that these descriptors can imply formatting manipulations far more numerous and complex than simply skipping a line and indenting. For example, ordered list items don't have to be numbered by the author; they are simply typed in with the :li. tag and can be re-ordered or otherwise manipulated, and the formatter will number them automatically. Similarly a large complex document with multiply embedded, and numbered, sections (e.g., "1.0 2.0 2.1, 2.11, 2.12, 2.121, 2.1211, . . .") can be outlined using certain heading tags which do not force the author to think about numbering or other matters at all;[7] when formatted, the headings are not only numbered correctly, but the text associated with the heading is also printed in the locally-preferred combinations of font-styles, font-sizes, and bold-face vs. plain.

The tags therefore provide means for freeing the user from thinking about how to format documents while generating the text, and there are many new schemes for simplifying the learning and use of such tags. The tags could also be considered to be structural landmarks in the text, and they might well assist authors by providing cues concerning the organization of the document. In fact, authors might even use tags — particularly heading tags — to outline a part of the document first and then fill in the section with text. As is discussed in the later section on planning, this idea is extended to provide for the computer to produce fully formed outlines for the author to follow, based upon the author's selection of a particular genre.

Additional aids

There are extraordinary opportunities for clever programmers, on the one hand, and educators with a feeling for exploiting computer technology, on the other, to recognize and act on the possibilities for new computer function for the scribing process. Below are a list of a few ideas (which can be found in some existing software in some cases):

Expanders. These can make it easier for people to refer to long or complicated phrases or character expressions as they work along, via short strings: thus .ibm., when entered, might cause the system to expand and enter the string to be "International Business Machines Corporation."

Phras-ers. This is the idea of breaking text into smaller meaningful units. If a person entered several sentences at one time, the code could intercept the keyboard input, find the ends of sentences, and put each sentence on a separate line. This code might also break sentences into various kinds of phrases, using simple phrase-finding algorithms, such that the subject of a sentence was on one line, the verb elements on another, and the direct object information on a third. These types of text separations might help an author see non-felicitous constructions or word-choices, or it might help in evaluating sentence complexity.

Revision marking. When portions of a text have been changed in a session (or previously), the altered sections could be indicated by changes in displayed color, margin indicators, or italics, etc. This might well assist an author in picking up the thread of a paper being worked on, and it could provide a useful record of editing activities.

On-line dictionaries. Many authors interrupt their writing to look up a word, to check its meaning or to discover a synonym or other type of substitute. It would be very nice to be able to access a dictionary just by putting the cursor on a word in the text and pressing some function key to have the definition appear on the screen.

On-line text/data retrieval. How often it happens (at least to this author) that in the midst of writing something one realizes one needs a name, or some piece of data, or some other example text to look at (and copy). How nice it would be, then, to be able to open, say, another window on the screen, peruse various other files until one finds what one needs, and then — presto — push the right buttons and have that information transferred into one's working file.

Speech input/output. Perhaps continuous speech recognizers won't be available for some while yet, but even small vocabulary isolated word recognizers would be very useful as input devices for text-editing commands and formatting controls. On the output side, thinking how useful it is often to read one's paper out loud, or to have it read to one, a high quality voice synthesized speech reading of one's text might be very nice.

COMPUTERS AND CRITIQUING

The second stage of composition that computers have entered extensively is that of critiquing, the stage following scribing. It is in this area, for the first time, that linguistic notions begin to be introduced into the computer software. While the degree of critiquing function that is now operational is quite broad, the possibilities are even greater, as indicated in the following section.

The space of possible critiques

The act of critiquing involves the examination of something relative to a set of standards followed by a part of the findings. In order to appreciate the rather vast possibilities of critiquing for compositions, five distinctions can be made: (1) the examination text-unit, (2) the report text-unit, (3) the critique type, (4) the strength of the critique report, and (5) the linguistic perspective of the critique. The levels of text we find generally useful to identify are: character, morpheme, word, token, phrase, clause, sentence, paragraph, section and document. A morpheme might occur as a stand-alone syllable in the text if, for example, words were hyphenated at the ends of lines; a token may be a word alone or a word combined with punctuation or other non-alphabetical characters (e.g., "and/or"), or it may be a word combined with a special formatting code, like the markup tags (e.g., ":cit.SGML:ecit."); clauses are identified as such only when there is more than one of them in a sentence; a section is a major portion of a document — like chapters in a book — and a document is the entire composition, however long, whatever the genre.

The examination text-unit refers to the unit of text which is examined for the presence of some target. If the critique is that of spelling-checking, then the examination text-unit is a word; if the critique is to determine whether numbers were used (instead of spelling quantities out in words) the unit is a character.

The report text-unit is the unit at which the critique report is made, and this unit is either the same as the examination unit or else larger. An example of the latter instance is when a text is critiqued for low frequency words (examination-unit = word) and the results are summarized on a paragraph basis (report-unit = paragraph), e.g.: "This paragraphs contains the following low frequency words . . .." Situations in which the report text-unit is larger than the examination text-unit are called aggregate critiquing.

The third distinction, critique type, refers to the manner in which the critique is made, and the two options are isolated vs. relative. In an isolated critique a particular examination-unit is compared against a standard, and the judgment can be rendered without taking into account the characteristics of that unit relative to other units. Thus, checking for spelling errors, incorrect capitalization, overly-long sentences, etc., all involve an isolated critique. In contrast, a relative critique checks the characteristics of one text-unit (having certain features) against the characteristics of another text-unit (having different features); the logic of the comparison is along the lines of "if the first unit has an aspect X, then the second unit must have an aspect Y." Most ungrammaticalities, such as disagreement in number between subject and verb, involve such a relative type of critique.

The fourth distinction concerns critique strength for which there are also two possibilities: right-wrong vs. threshold. A right-wrong judgment is one in which one can say "Right!" or "Wrong!" without fear of contradiction (from experts), as in the case of the majority of grammatical errors. Thus, "two apple" and "He aren't" are just plain wrong. On the other hand questions of

style are not only matters of taste but also the province of the organizational enterprise authorizing the composition; one must be very careful in critiquing these things, since observations like "the sentence is very long" or "the structure is not parallel" aren't sufficient for strong criticism in themselves. Stylistic problems must typically needs be reported with some deference and sensitivity to the fact that the author and critiquer may not share the same standards. One means of systematically handling the problem of varying stylistic standards is to arrange to have each stylistic evaluation result in the computation of a single number whose value grows with the severity of that particular gaffe; this value can then be compared against the threshold for a particular enterprise, and, if it exceeds that threshold, a suitable commentary is provided.

The fifth factor, the linguistic perspective, invokes the system of semiotic theory (cf. Morris 1946), which identifies three levels of linguistic consideration: syntactic, semantic, and pragmatic.

Such a large multi-dimensional critique space is rather unwieldy in practice, especially since the development of language technologies is really in its infancy. Accordingly, we have organized the EPISTLE critiquing function into three types: word, grammar, and style. Word level critiquing now includes consideration of syntactic problems (misspellings) and someday will be extended to semantic problems (e.g., inappropriate word-choices — using "infer" where "imply is intended). Grammar critiquing focuses on sentence-level phenomena, and style can be discussed from the word-level up through the whole document.

Word critiquing

The most prolific area of critiquing is that of words, especially for detecting errors in spelling. Recently, some linguistic competence has been introduced into the computer software, in the form of some knowledge about "words" — although this knowledge may be quite primitive and consist only of a list of character strings: anything found to be on the list is considered to be a word; anything not found on the list is a spelling "error." Despite the low level of linguistic knowledge about words possessed by computer spell-checkers, there are some remarkable successes, and the function delivered is surely valuable (e.g., Durham et. al. 1983; Pollock & Zamora 1984). The function has been augmented in other products, including IBM's PROOF product, to include detection of awkward phrases and the capability to suggest synonyms (e.g., Leerburger 1981; IBM 1983).

The most extensive dictionary capability function appears to be that provided by the IBM Yorktown Watson Research Center's experimental EPISTLE system, whose on-line dictionary provides strings of syntactic and semantic features, for use by the grammar component, for some 130K words (cf. Byrd 1983, 1984; Byrd and McCord 1985).[8] Perhaps half again as many words are recognizable by use of the extensive capability of this system to analyze an unknown word into and deduce its part of speech and other characteristics (so-called "inflectional and derivational morphology").

Word critiques which involve reporting of word problems at the sentence level include (1) incorrect spellings, (2) non-preferred spellings, (3) incorrect abbreviations, and (4) incorrect capitalization (e.g., "dr."). These four sentence-level critiques are also critiques which are triggered by the simple presence of the problem — occurrence-level targets. A sentence-level examination critique involving an aggregate-level report is (5) one which checks for repetitions of substantive words in the same sentence (e.g., "The cool light was cool on my face.").

Other possibilities for word-critiquing at the discourse level include five main classes: (1) observations on the relative frequency of word usage from various types of dictionary sources, (2) observations on tonal features of the words, (3) detection of hackneyed words or phrases, (4) detection of incorrect or inappropriate word usage, and (5) comments on relatively subtle changes in usage of words, particularly the same words, from one part of the text to another. The first two categories involve aggregate reports on the text, still at the sentence level of detection; the remaining three categories involve occurrence reporting at the sentence, sentence, and discourse levels, respectively. Categories (4) and (5), in particular, require much more extensive linguistic processing than the simple dictionary look-up of the first three.

The type of word-critiquing provided by (1) above would inform authors how many words in their composition came from "everyday" vocabularies, technical jargon lexicons, or — heaven forbid — special term glossaries. That of (2) would reflect the overall "tone" of the document in terms of such features as formal vs. informal, or colloquial/regional/slang, or neutral vs. evaluative, or friendly vs. unfriendly.[9] A report might indicate that the text, although quite informal overall, did contain several areas where excessively formal words were used. Category (3) involves the search for overused, cliche-type expressions which are more or less "fixed." For example, in the clause "We deem it advisable . . .," the key worn-out words are those in italics; one would hope to identify them no matter what variations of other words in and around them were made.[10] All three of these types of word critiquing are planned for the EPISTLE system.

While the first three types could be along quite well with relatively simple dictionaries and little further linguistic processing capability, the last two would require not only syntactic analysis of the text, but also very capable semantic processing. What is envisioned is the capability to provide the following types of diagnostics: "You probably meant to use the word <'infer'> here instead of <'imply'>" (class 4); "This word <WORD> simply doesn't make sense here" (class 4); "You are using the word <WORD> in two different senses in the same context" (class 5); or "Your use of the word <WORD> here implies one thing, but the use of the word <WORD> in the next sentence implies something very different" (class 5). Word usage critiquing of this type is well beyond the capability of any existing system.

Grammar critiquing

The notion of grammaticality is almost entirely restricted to a sentence unit. One doesn't have "ungrammatical" words — only words that are misspelled, oddly spelled, or, morphologically speaking, strangely formed. Similarly, paragraphs are text sections that may be called contentless, poorly organized, or highly redundant, but they are not deemed to be grammatically incorrect. The major class of sentence ungrammaticalities is that involving agreement — agreement in number, person, gender, and case. While English is one of the least inflected of languages, there are still a number of opportunities for grammatical disagreements.[11]

The detection of failures of agreements and other ungrammaticalities in a sentence requires that the sentence be "parsed" according to the grammatical rules of the language, resulting in a description of the syntactic structure of the sentence. This description must not only tell about what modifies what in the sentence, but also about the grammatical roles played by different parts — e.g., the subject, the main verb, the direct object. Parsing quality can be judged in two ways, both related to the adequacy of the syntactic descriptions. First, the modification structure can be judged as to its accuracy, the most important aspect being "were the words in the sentence assigned the correct parts of speech?" For example, the sentence (1)

(1) Time flies like an arrow

can be understood quite easily in the sense of (2)

(2) This airplane flies like a rocket

fairly easily in the sense of (3)

(3) The fruit flies do like an apple when they can get it

or with some difficulty in the non-obvious but still acceptable sense of (4)

(4) Please time the athletes in the manner of an Olympic judge

In the sense of (2) the word "time" is used as a noun; in the sense of (3) it is used as an adjective; and in the sense of (4) it is used as an imperative verb. For this example all three readings are possible, and the "correct" one is decidable only from other information. In these cases of true constituent ambiguity the structural descriptions provided by the parser will differ greatly; the follow-on processes which use these descriptions — critiquing, for example — will also vary widely. Thus, without complete understanding of the whole text, it is possible to make a reasonable misunderstanding of a single sentence.

Other problems involve the notion of "attachment." For example, in (5)

(5) I saw the man in the street

the prepositional phrase PP "in the street" either could specify which man or it could refer to the location where the "seeing" action took place. Therefore, the PP could be attached to either the noun phrase, NP, or the verb phrase, VP; in this example either could be argued. The question, though, is the general trend of attachments over a large number of parses: what is the safest course, and what are the various consequences? The IBM EPISTLE grammar approach, in the first version called here EPISTLE-1,[12] was to attach PPs to higher-level VPs

rather than nearby NPs — unless other local information indicated otherwise (e.g., PPs with the preposition "of" were often attached to the immediately preceding NP). This approach, while not necessarily more correct in the long run than some other similar decision did have the consequence of "raising" these phrases in the parse tree, making them more easily examined. This kind of bias would lead to a certainly correct attachment of the PP for (6),

(6) I saw the man through the binoculars

a possibly correct attachment for (7)

(7) I saw the man with the binoculars

and a probably incorrect attachment for (8)

(8) I saw the man with the raincoat.

The second general way in which parser/grammars can be evaluated is by the accuracy with which they determine the grammatical function of various components of the sentence. Thus, an adequate parser in this sense would identify "John" as the "real" or so-called "deep" subject of (9) and (10) despite variations in the surface subject:

(9) John hit the ball

(10) The ball was hit by John

A functionally-accurate parser would also correctly identify the indirect objects in (11)-(14) as "him," "her," "whom," and "who" respectively:

(11) She gave him the book

(12) He gave the book to her

(13) To whom did she give the book?

(14) Who did he give the book to?

In addition to quality or accuracy, parsers can be judged also on their "robustness" in the face of ill-formed input text: does the parser wade through "noisy" errors and come up with something; or does it throw up its virtual hands and come to a screeching halt at the slightest difficulty? And, in particular, what does the parser do with sentence fragments? For these problems the original bottom-up EPISTLE-1 parser is also quite satisfactory, having several different strategies for handling — and surviving — problematic text, including unrecognized words, sentence fragments, sentence run-ons, ungrammaticalities, and the grammatically correct but troublesome instances of sentences with multiple legitimate (and illegitimate) parses (cf. Miller 1982a; Jensen & Heidorn 1982; Jensen et al. 1984). For the new top-down version of the EPISTLE parser now being implemented, called here EPISTLE-2,[13] there was initial concern that it would be quite difficult to use it for describing the partial and excess structures of sentence fragments and run-ons, respectively. However, it now appears that the same grammar can be used to accomplish these identifications with no re-parsing.[14]

Given that one has a parser, then — and there may be several available — the interesting question from the point of view of composition is the extent to which the parser can be used to supply grammaticality information about the

text to the author. So far as we know, the IBM EPISTLE-1 prototype is the only one providing such function, and it provides for detection of 27 specific classes of errors, shown in Figure 3.

While the performance of the grammar can be enhanced, there is really not that much more that can be added to this set of grammatical critiquing targets. The practical problems remaining concern: (1) improvement in grammar performance,[15] (2) extension of the grammar-checking capability to new text domains involving different sublanguages and grammatical conventions, and (3) development of grammar components for other languages.

Style critiquing

In the critiquing of style we leave the firm safe shores of grammar absolute and push out to the seas of relativity: "This is perhaps awkward here"; "If your point is thus and so then it doesn't really come across"; "Do you really mean to say that?"; "Must you include this section?"; "I don't understand you here." Where there is more or less one single burning flame of grammatical correctness, there are hundreds of flickering candles of style, each attesting to the eternal truth of *"chacun a son gout"!*

Style is in the eye of the beholder; more precisely, style is the expression of an author's strategy for communicating information. For the texts of commerce, authors don't have to worry much about coming up with these strategies; their enterprise, their communication medium, and their field all act together to provide more or less explicit guides for style (as discussed in detail in the next section). Even more available is the capability to detect deviations from the "good" style of an author's place of business: either there's a "style handbook" lying around or else there's always an eager old hand who's quite willing to point out the numerous flaws of the stuff of the newcomer who has yet to learn the ropes about "how we write things around here."

Many people have initially denied the truth of this assertion, but, upon questioning, they all have had to concede that, yes, for every type of business or technical writing they did, they were either following the dictates of some crusty mental guide or else they could recall who or what was their source for good style when they had questions. For example, a business person will respond to a memorandum with a reply containing the same format, fields, and breadth of content; the quarterly sales report will be stylistically identical to that of the previous quarter; the notes to colleagues, the "instructions in my absence," the technical review, and the personnel appraisal all similarly will reflect thoroughly the accepted stylistics of that environment. Thus, if the first point made about style was variability, then the second point is quality control! Each enterprise sets the standards — as it sees fit — for controlling the form and conformity of each type of writing output. For better or worse, each contributor to the writing flow has to internalize these standards.

The contribution of computer software to this situation is the capability to provide each author a private forewarning of stylistic criticisms that might be delivered later from human agents. The computer critiques can be a much more agreeable situation, particularly when suggestions and explanations are

1. Subject-verb number disagreement ("The leaves falls")
2. Noun-premodifier number disagreement ("Each valves . . .")
3. Subject-complement number disagreement ("This is the books")
4. Subject-verb person disagreement ("I likes to eat")
5. Confusion of "it's" and "its" (It's nose is dirty")
6. Wrong pronoun in predicate nominative ("The right person is her")
7. Wrong pronoun in object position ("Between you and I, it's a cinch")
8. Wrong pronoun as indirect object ("I will give they the book")
9. Confusion of "who's" and "whose" ("Who's side are you on?"
10. Confusion of possessive 's and plural s ("Few pet's are lonely")
11. Confusion of semicolon and comma ("Please buy coffee; milk; and eggs")
12. Conjoined verb number disagreement ("The sand both glitters and choke")
13. Incorrect double modals ("'He should ought to go")
14. Existential "there" number disagreement ("There's five apples here")
15. Improper form of the infinitive ("He wants to won")
16. Improper form of the verb ("She broked my heart")
17. Missing obligatory commas ("He decided I think on the plan")
18. Missing obligatory hyphens ("I have a yet to be satisfied request")
19. Missing an obligatory question mark ("What is your name.")
20. Mismatching correlative conjunctions ("You are neither happy or sad")
21. Mismatch of the number of a noun phrase and its relative clause ("I saw the man who are here")
23. Sentence fragment or "non-sentence" ("Also my tires")
24. Unbalanced comma ("My manager, Mr. Hendrix told me about you")
25. Unmatched pronouns ("Him and I are going")
26. Use of wrong indefinite article ("I ate a apple and an pear")
27. Use of "of" for "have" ("I could of danced all night!")

Figure 3. Grammaticality critiques

given for each detected violation — as they are, with adjustable levels of detail, in the EPISTLE-1 system. For example, were a person to have typed the sentence (15)

(15) Between you and I, I am sure she will succeed.

the system, when asked to critique, would first underline the word "I" in red (the usual editing display is green text on a black background). If the person pressed the appropriate HELP button, a box within the text-editing screen would open up and a two-part (colored, reverse-video) message would appear saying: (1) Wrong pronoun in object position (prussian blue); (2) me (aqua marine; this strategy of (1) classifying the problem, and (2) suggesting something to do about it is followed for all three types of critiquing in the system). But then, a second press of the HELP button produces an extension of the message, a 20-60 word explanation of the tense classification; and a third press produces a detailed tutorial on the general topic involved, from half a page to

three full pages in length. Thus, the author can not only locate each problem but can obtain replacement suggestions and increasingly detailed explanatory information in every case.

If such critiquing can help the peace of mind of the writer — and actually promote better writing skills — this is certainly of benefit to the enterprise, from the point of view of the satisfaction and competence of its employees. But, setting personnel concerns aside, such capability can provide the enterprise even more direct and tangible benefits by assuring that its written output is of constant high quality, according to its own dictates. Thus, at least proposals won't get dismissed because of poor grammar or awkward style, a newspaper will avoid being a laughing-stock because of a misspelling or tortured expression, a publisher's reputation for quality won't be smirched by missed "the . . . the"s (at the end of one line and the beginning of the next), and a manufacturer can have confidence that his assembly instructions are readable.

Where there are only a few dozen grammatical error possibilities at most, the number of stylistic targets — per individual enterprise — can easily be in the hundreds, and each major commercial area (e.g., education, legal, manufacturing, etc.) may have thousands, with only a few of these overlapping other areas. For example, the stylistics of legal writing are considerably different from all other types, but within this area, the specifics will vary greatly from one state of the union to the next, and even from one federal government agency to the next.

The consequence of this great diversity of styles is that a system purporting to provide maximally useful stylistic critiques must have been customized to the major commercial area, to the major subdivisions within that area, to the requirements of individual enterprises within these subdivisions, and even to departments within enterprises. We would further hold that every individual user must have some control over the stylistic targets — either to turn them on or off or to adjust their level of sensitivity.

EPISTLE-1 sentence-level style critiques

Many of the prototype's critiques are of the threshold type, in that the "strength" of the problem is computed on an increasing numerical scale, and this strength is compared to some threshold value; when the strength exceeds the threshold, the critique is voiced; when the threshold is not exceeded the system can quite sensibly keep its "mouth shut." These critiques, as well as the ones that are based on the direct occurrence of a pattern of present/absent symptoms, are far from being simplistic, in that the context is examined carefully before criticism — the same construction might be quite acceptable in one context but simply awful in another; the style critiques attempt such sophisticated discrimination.[16]

Almost the sole information source for the stylistic computations is the annotated parse tree which is output by the grammar component. This tree details the syntactic structure of the surface sentence, assigning grammatical roles to some elements (e.g., subject, direct object), providing information on all of the inflectional aspects and giving the semantic features associated with

each text unit. The stylistic rules interrogate this tree, measuring it, counting aspects, comparing and contrasting, and develop the specific features needed for each critique target. Thus, the style component performs no additional general processing of the sentence or its parse-tree representation; measurements are obtained according to specific target descriptions directly from the tree.

The nature of the stylistic processing means that once the parse-tree has been obtained, the hard part is over; the rest derives from examination of that tree. The implication of this approach for obtaining new style targets is that it will be a relatively easy thing to accomplish — as long as the features that reveal the targets are derivable from the tree. Thus, we anticipate that it will be a relatively straight-forward matter for a specialist to implement new capabilities.

The source of most of the stylistic critique targets presently implemented is the general consensus that exists concerning "good" and "bad" style (see references in Miller 1982b). These will probably be of interest no matter what the details are of a particular enterprise or a user within that enterprise. However, each enterprise will have many specific "bad-style" targets, not included in the general set, and these will have to be custom developed. We estimate that perhaps 70% of an enterprise's special targets can be implemented to a reasonable degree by using information from a parse tree. The remainder (e.g., "Does each topic logically flow?", "Is it well organized?", "Is it informative and precise?") will be impossible to implement at the present time, because of two problems: (1) the critique requires complete semantic interpretation of the sentence, probably involving knowledge-based inferencing, and (2) the critique requires interpretation of large chunks of text, more than one sentence at a time — so-called discourse-level analyses. In Figure 4 are listed a sample of the 106 different style critiques presently implemented as EPISTLE-1 system function. Note that some punctuation errors are included; many such errors will rely on very extensive semantics for their detection and will not be achieved for some time.

1. Sentence too long (or paragraph, procedure step, or heading)
2. Sentence is not parallel
3. Too many nouns modifying the head noun
4. Too many consecutive prepositional phrases
5. Commas might be missing around non-restrictive modification
6. Preposition and participial are too far apart
7. Incomplete verb construction
8. Incorrect use of "like" as a subordinator
9. Incorrect use of comma as a complementizer
10. Possible excessive use of "and"
11. Incorrect comma between conjoined phrases
12. Possible excessive use of negation
13. Too many hedges/concessions
14. Overused phrase

Figure 4. A sample of EPISTLE-1 style critiques

The difficulty of writing these very general style critiques was compounded greatly by the requirement that the criticisms be reasonable over a wide variety of texts and authors, from college to the military, and from student to chief editor. We expect that it will be relatively easy to develop new critiques for a particular enterprise, especially one that is extremely "picky" and precise about its style criteria (such as a publisher).

Aggregate Style Critiquing

There are at present very few examples in EPISTLE-1 of this type of critique target, primarily because the others were more challenging and pressing to develop. We will include in EPISTLE-2, however, a number of aggregate critiques, summarized over larger text units of the document. For larger documents especially, we envision one form of the output of aggregate reports as being a kind of graph, with the horizontal axis marking off chapters, sections within each chapter, paragraphs within each section, etc. For one or more of these text units them, frequency or percentage information on the following kinds of targets could be provided (the list is meant to be illustrative and is not exhaustive of the reasonable possibilities):

- Information Density
  - Words
  - Sentences
  - Facts[17]
  - References[18]
  - Specialized document entities[19]
- Word Features
  - Tonal aspects (formal, colloquial, etc.)
  - Use of rare or archaic words
  - Use of jargon or technical terms
  - Use of abbreviations or acronyms
- Sentence Features
  - Type (declarative et al.)
  - Voice (active vs. passive)
  - Complexity (simple, compound, et al.)
  - Readability

Discourse-level critiquing

The real aspects one wants to check in a document have to do with how effective it is in achieving its goals. Sentence-level style and diction are important, of course, but they do not begin to assess the quality of the overall document. At the discourse-level one certainly is interested in at least the following: Specific content, organization, coherence, relevance to topic, information-value, clarity. None of these can be assessed by computer software at the present time. And one pales at the thought of how much science and art have yet to be invented to make computer critiquing of this kind possible.

There is one possibility, however, for making some headway sooner, if authors will agree to some extra work. The idea is to use the tagging approach, discussed in the section on scribing and formatting, for authors to annotate their running text in terms of the rhetorical functions it is intended to achieve. The following is an imagined list of what a few of these tags might be:

:kprop. Key proposition or argument to be supported

:prem. Premise or assumption

:contx. Background or context of discussion

:tstruc refid=T1. Topic structure to be introduced, e.g., a set of 6 items to be discussed in turn; an element of this structure may refer back to this by using the reference identifier "T1."

:next ref=T1. Next item in the list spelled out by the topic structure identified as "T1."

Were an enriched and more carefully developed set of such tags developed, the analysis capability of, say, EPISTLE-2, would make it much more feasible to evaluate the free text enclosed by the tags in terms of the criteria listed above. It is not at all clear, however, whether authors could be so analytical as to identify the rhetorical functions of each text segment, nor whether they would tolerate having to use tags even if they knew. The point is that a compromise such as this does move forward the functional capability to do discourse-level critiquing — to perhaps just a few years away rather than a decade or more.

Format critiquing

A final type of critiquing concerns how the words are arranged on the paper (or computer screen), relative to how they should be arranged according to the writing-genre employed and the local customs. For example, it may be the policy of an enterprise for the title page of all technical reports to have a security classification as a running bottom margin item, report title centered ⅓ down from the top, the author's name followed by two blank lines and the address, etc.

The format critiquer should verify that all these formatting standards are met, first by identifying what type of document is being written, and then by checking the document against each of the standards. Since authors may not identify the document as being a memo, a report, a letter, or whatever, the system will have to have the capability to infer the genre — either from the embedded tags (if the source file is submitted for critiquing) or from the arrangement of blank space on the page (if the formatted file is submitted).

There's not much help in this area yet, but the requirements are clear, and implementation really depends more on standardization of document types and tags than upon the invention of new algorithms.

Commercially available systems

There are several commercially available systems which do provide some level of text-critiquing function. Perhaps the most well known of these is *Writer's Workbench,* which is marketed by Western Electric (cf. Cherry et al. 1983). The

program essentially comprises four functional components: proofreading (spelling, punctuation, wordy phrases), style analysis (sentence length listing, distribution of sentences by type and complexity, sentence voice), information (English tutorials), and utilities (personalized dictionaries, new style standards). An extensive review of this product's function and performance is provided in Seybold Report (Seybold 1984); this report also reviews imitations of Workbench, the EPISTLE-1 system, and a number of other similar products.

The advantages of these products is that they are available now at rather reasonable costs, and they usually can be run on a stand-alone personal computer. The disadvantages may be that the function provided is necessarily limited by the size and development of the underlying language technologies — (1) the dictionary and (2) the parser and grammar; in most cases the dictionary contains only a few hundred function words, and the "parsing" is pretty much probabilistic guesses at parts-of-speech and identification of phrases — but nothing like a detailed description of the hierarchical syntactic structures.

In addition to the above limitations, there is no provision in existing commercial systems for accomplishing any semantic processing or knowledge-checking. Such capabilities not only will probably require large storage (a megabyte does not seem unreasonable), but also this type of processing can be quite involved and can occupy large amounts of CPU time. Extensions of function into these areas will most likely await the availability of much much faster and larger personal computers.

COMPUTERS AND THE REMAINING COMPOSITION STAGES

Whereas the previous discussions of scribing and critiquing have largely been focussed on actual software systems, the discussion of the remaining stages is largely speculative: systems to accomplish most of the described function have not yet been built. Nevertheless, the development of computer support for the first three stages would appear to be a somewhat easier task than, certainly, the development of large on-line dictionaries and parsers.

Computers and communication requirements (stage 1)

The key problem in this stage is determining what goals the text should meet; before these goals are firmly established there should be, ideally, no consideration at all of how the goals are actually going to be met. This phase actually occurs with familiar regularity in large text projects, particularly commercial ones. Thus, a book publisher will go through this phase in considering development of new titles, reference publishers will do so in considering the merits of issuing a new collegiate or learner's dictionary or junior encyclopedia, and technical journal editors will do so in thinking of a special issue on a particular topic. In all of these cases there will be careful consideration of the communication goals, the expected audience, the potential delivery media, the specific content to be covered, the various contexts to be related to, and all sorts

of incidentals like who is it that has to be flattered by all this, and what are the possible details of schedule, pricing, resources, and follow-on needs.

But what of the lowly interoffice memo, a weekly paper in English Comp, or an outraged complaint letter? Is the communication requirements phase truly a useful aspect of these much more minor pieces? Since these all still have rich structure and pragmatics, the answer must be, yes. However, whereas the really big projects often have access to detailed guidelines for determining functional requirements, these are much less available for the genres in which the output is so much smaller and tidier. Ironically, it is these smaller genres, and not the larger, which have received some initial computerized support, albeit highly formulaic (cf. Schwartz and Bridwell 1984). One of these, WANDAH (Writing Aid and Author's Helper; cf. Von Blum and Cohen 1983), provides this component in the context of several other aids, including a text editor. An advantage of such programs is, of course, that one can begin to see the potential for true support of composition throughout the whole process. The difficulty is that these programs have no way of determining whether students' responses to their prompts are in any way reasonable — or even relevant.

Ideally, computer support of this phase would involve three aspects. First, the computer would present the would-be author with a set of "considerata" to reflect upon and respond to. These would be derived from, and ordered in accordance with, a detailed psychological model of this very early stage of composition. Second, the computer would assess the quality (detail, clarity, relevance, etc.) of the author's response to the "considerata"; were the quality insufficient, the computer could provide feedback (e.g., prompt, chide, fret, or bully). Third, and by far the most difficult, the computer could assess the product of later stages of composition in terms of how well they implemented the requirements; detection of incompleteness or contradiction could trigger bringing the author back to consider again the functional goals set forth way back when, to resolve the discrepancies. None of these three things is likely to happen for a while, since there's nowhere near the necessary and proven cognitive theory to support such work. Even if the theory were available, there's hardly a clue how to achieve implementations of this kind.

Computers and high-level planning (stage 2)

In this stage the author leaves consideration of what should be achieved and ponders the how of it. For example, suppose the communication requirement is mainly to get back at some agent of commerce who refused to replace or credit a defective piece of merchandise. It's not at all certain that the communication action must involve a letter to someone in that enterprise; perhaps, instead, an ad should be taken in the *Times* decrying the action; or perhaps a telegram should be sent to a friendly and influential senator. The decision to be taken at this stage is to to develop a broad general approach to the problem, after considering a broad set of different approach concepts. If everything that one has thought about in this stage is but a slight shade off from one another, then likely one has not pondered long — nor "High" — enough.

If there is doubt about being able to support the prior stage with computers, there is at least as much uncertainty about it for this stage. What is really at stake here is the act of developing an effective overall plan of action from a set of goals. What is needed is a pragmatic handbook which gives details of procedure, and evidence of success, for the cornucopia of possible communication plans and approaches matched to goals. Were this guide available, it could most certainly be converted for on-line use. Less certain, of course, is the extent to which authors' communication requirements statements can be evaluated against subsequent plans to accomplish the stated goals.

Barring such a handbook, there's not very much that can be computer-implemented at this point except for very general open-loop guides. There are in fact several software packages of this sort which are inteded to help the writer develop a working plan which moves easily into the actual scribing task (see Seybold 1984; Schwartz and Bridwell 1984; Nancarrow, Ross, and Bridwell 1982; Dalgish 1984). Two limitations with these programs are that they are (1) very general and undifferentiated with respect to specific writing topics, and (2) they have no way of evaluating the students' input.

Computers and detailed planning (stage 3)

At this point the author has narrowed the possible approaches to a single major one and now must ponder the details: e.g., "A letter, surely, but how phrased? . . . and could I be sued for libel?" Since computer support of design has been adequately reviewed in the previous section, and since no one else seems to differentiate between high and detailed design, what is given here is a proposal for providing rather strong and closed-loop detailed design support in a manner which moves quite easily into the following stage of scribing (and even critiquing).

In the absence of data to the contrary, it is useful to presume that one of the key decisions of the high-level design was to decide upon the major type of communication genre to be employed. That is, you know roughly whether you have to write a report, a letter, a legal document, a journalistic article, or a fictional story. The essence of the present proposal is to have the computer provide you with a way to narrow down the broad set of choices that still remain until you have chosen a quite specific document genre, and then to be able to tell you all the details of the pragmatics of this document so that you can start adapting it to your particular needs.

The first key element is the detailed classification of genres within an area, organized in such a way that writers can find their way to a specific sub-genre which will meet their needs. Considerable thought was given to an approach towards classifying major genre areas to minimize overlap and to reflect human pragmatics strongly, and the choice was to partition major genre branches according to the major social institutions: law, medicine, education, military, the press, government, commerce, etc. Two of these branches, law and the press, were chosen to be developed, to get some feeling for the proposal.[20]

The first task was to develop a workable taxonomy of writing genres within each field, organized from the viewpoint of the demands on an author to write within a particular genre. About 250 sub-genres were decided upon for both law and journalism, and the first few levels for law are shown in Figure 5 (Halpern 1984a).

This taxonomy (and the one for Journalism as well) was developed and tested iteratively by repeated examination of (1) diverse textbooks and (2) many actual examples. The development of the finest genre distinctions, the sorting of these in the hierarchical tree, and the naming of the classification nodes themselves, all continued until new input was easily identified and caused no problem in classification. These sub-genres were also cross-classified in terms of their values for a number of independent dimensions relating to their rhetorical quality. A variety of different dimensions were considered, and those that were retained (1) had some face validity with respect to rhetoric and communication and (2) showed a reasonable variation in values across the several hundred sub-genres. The dimensions settled on for Law are shown in Figure 6 (Halpern and Miller 1984, 13-14).

With the taxonomy and the dimensional classification systems then, the first part of the proposal was accomplished: Authors, knowing that the communication objective required writing a legal (or journalistic) document, could find their way to an extremely specific sub-genre which could accomplish their purposes. The second step is to provide the computer support to assist the authors in understanding all aspects of this genre, and to provide the framework for moving easily into scribing from this stage for elements of the document, and then back again into detailed planning.

The means for supporting the second step was to permit an author to choose a sub-genre, be put into an editor showing the high-level skeleton of that genre, and then be able to inquire about three aspects of any part of that skeleton: (1) the sub-classification, (2) the writing conventions, and (3) the rhetorical functions. Figure 7 shows the top level structure of a dwelling lease, in which an author has asked for elaboration of the property aspect, and then, within that the facilities included (Halpern 1984b).

By positioning the cursor on a line and pressing the appropriate key, an author can accomplish the three functions desired, that of expanding the category into its next levels shown in the figure. To determine the rhetorical importance of asking about the garage, for example, the author would position the cursor on that line, and press the *Rhetorical Purpose?* key; in this example, there would be a short response pointing out that garages are often contracted for separately and are often a basis for dispute if not specified clearly, etc. Pressing the *Writing Convention?* key might produce information that the key aspects of the garage to be covered are (1) its deviation for "normal" size (perhaps can accommodate only compacts), (2) number of bays, (3) whether it opens on a street or in a private area, and (4) whether there is room for storage. If this aspect called for special formatting, use of capitalization, or other non-textual aspect, that would be presented also.

1. Legal argument / scholarship
   a. Motion / petition
   b. Brief / correspondence / opinion
2. Records
   a. Affidavit / bill of evidence
   b. Deposition / record / minutes
   c. . . .
3. Pleading, orders
   a. Summons / warrant / citation
   b. Decree / order / injunction
   c. . . .
4. Legal transactions
   a. Free form (contract, offer, convenant, will, . . .)
   b. Fixed form (bond, stock, offering, order, . . .)
   c. Non-common law
      1) Legislative (bill / statute / ordinance)
      2) Constitutional (amendment / referendum)
      3) Admin / executive (treaty / regulation)

Figure 5. Taxonomy of legal writing genres

1. Density of promissory statements
2. Density of opinions
3. Extent to which authority is cited
4. Relative mix of sentence types (declarative et al.)
5. Density of boilerplate language
6. Extent to which the form is prescribed
7. Density of terms of art
8. Density of abstraction
9. Characteristic expressions and terms of art

Figure 6. Dimensions of legal style

1. Document title
2. Date of this document
3. Parties
4. Description of property
   a. Address
   b. Facilities included
      1) Number of rooms
      2) Description of each room
      3) Whether garage is included
   c. Period of lease
6. . . . (and 16 additional sections)

Figure 7. Structure of a dwelling lease

The usefulness of this approach from the detailed design point of view is that authors can be helped to rapidly identify and learn about the sub-genre that will meet their communication goals. In addition, however, it is easy to see how an author can keep right on going into the scribing phase: having seen the lowest category level (e.g., "whether garage is included"), it would be quite easy for the author to simply generate appropriate text for that section, according to the rhetorical function and writing convention information supplied. Then, having written something, the critiquing phase can be called upon to use the detailed tags, and all the other associated information, to assess whether what was written was relevant, coherent, effective, etc. — all those impossible discourse-level aspects discussed previously.

While this approach has yet to be implemented and data on its utility obtained, still it does provide a constant high level of support for the composition process through most of its important phases. Whether this particular concept is the best or not is of little consequence relative to the possibility raised here of more effective computer support — and once again through the use of identification tags.

AFTERWORD

After such a trek through the composition process, it is good to reach a vantage point and consider longer views. We shall take two, one looking at other audiences, and the other looking at the roles educators can play in this, their, new computer future.

Specialized audiences

The intended audience for the various computer aids was never really discussed in the foregoing, although technical writers were surely a focus. But nothing said so far really precludes the traditional educational audiences that are briefly considered here.

College. A phenomenal upsurge of interest in composition, particularly computer-based, has occurred in the last few years. New sub-departments — even departments — of rhetoric are formed (or dusted off), linguists are added or adjoined to the English faculties, and there's tremendous interest in the other Pascal and things like operating systems, non-human editors, RAMs, and bytes. Of all of the educational areas, it is in the colleges and universities that there is the greatest interest in computers and composition, and perhaps the greatest capability to pay and staff such activities. Our experience in examining college compositions and attempting to run them through our EPISTLE-1 system suggests that such an interest may be well-motivated: there is certainly more, in the area of style and diction, for these students to learn than might have been expected. Clearly, there is considerable demand for particularly the critiquing capability, but considerable modifications will have to be made to existing technology to survive the hardships that some of the less well-trained students will place on the system.[21]

9-12. High school students may well be the most amenable to working with the guiding function concept described in the section on computers and detailed planning.

K-8. The underlying language technologies, if properly interfaced, could well provide exceptional opportunities for training the language newcomers in the basic skills of this craft. For example, were a large dictionary to be accessible in a variety of interesting ways, children could, greatly through self-discovery, learn that words are composed of parts (some fairly standard), that words which look different might sound alike (and vice versa), and that the meaning of words provides a rich set of connections among them (some positive, some negative, some generalizing, some specializing). Similarly, the hidden non-linear structure of a linear sentence could be revealed, showing parts of speech, phrases, and grammatical function, all changing as this or that part is changed by the student.

Remedial. The computer offers special promise for those needing extra assistance because their needs can so often be precisely determined and prepared for. A side benefit, but one which may well greatly increase the will to learn, is that the help, discovery, and mastery can occur in a private dialogue between student and terminal.

ESL. English is a second language for millions of people in this country, only a few of which are adequately enrolled in the school systems. The rest must make do with inadequate night training, books, and the help of friends, and, of course, their learning is slow, limiting their potential enculturation, assimilation, and opportunity for better employment. It may be that a number of the errors a non-English speaker makes can be predicted from the natural tongue (the remainder are developmental and also well known), and therefore specialized critiques can be developed in a rather straight forward manner. The potential benefits to this population, then, are enormous. The other aspect of ESL is that of foreigners, quite happily living in their own country, but with a need to write in English. Since poor English can be such a stigma, causing potential buyers to look elsewhere, the commercial possibilities associated with this opportunity are very great.

Role of the educators

The last item of concern is the potential reaction of those who must ultimately midwife the computer technology of composition. When faced with the support capability discussed here, educators and other administrators can adopt a variety of positions, fairly well covered by the following four: (1) they can ignore the whole movement; (2) they can oppose it; (3) they can evaluate it; and (4) they can shape it.

The first position is unlikely to be a quiet one, what with students clamoring for word-processing and more, the school administration pushing for greater "productivity," and parents going on about higher quality education. The second position of actively opposing these trends, while certain to make the front

pages for a while, is much more dangerous if, as seems very likely, this capability is not going to go away, but grow rapidly stronger. Determined opposition may very likely leave one arguing at the backs of the retreating educational mass. The third position of testing the technology is certainly very constructive and absolutely necessary to some degree, if only to keep the salesmen's claims honest. The problems are that this is (1) awfully costly in terms of money, time, and personnel, and (2) it may well be quite premature, since the field is changing so radically. Only the very flexible and fleet of well-heeled foot can really succeed in this post.

The last position is one in which every one can participate, but is the least popular at this time. Nevertheless, the development of effective computer-based writing tools requires the melding of the two disciplines: the computer-linguist and the educator. Neither can hope to do a truly reasonable job without the other, but it is the computer developers that won't let a little thing like that stop them! What the educator can do, then, is to understand better the nature and limitations of the technology and then help the developer bend his wares towards those areas in which the most gain can be made.

One rather painless way an educator can provide useful stimulus is to think practically about what kind of specific writing function is needed most. These ideas can then serve as functional requirements to the computer linguist to design and build more effective and useful systems — sooner. To put it more succinctly: *Educators: Get involved, Now!*

1. A more complete model usually has a "problem formulation" stage preceding the first stage of functional requirements, an expansion of the fifth stage of testing into three stages ("unit test,", "system integration," and "system test"), and an addition of another stage somewhere after installation — "Maintenance").

2. Saying that such-and-such activity occurs in this or that stage is not an observational generalization. However there is substantial evidence, not easily dismissed, which suggests that design processes — whether of text or of programs — tend to have more desirable properties when, in fact, the activities and the stages are matched as described (e.g., Malhotra et al. 1980; Carroll et al. 1980). In this sense, then the model is a normative rather than a descriptive one.

3. In earlier polyglot times an additional decision would also be approached — that of the language to be chosen (e.g., French for matters of philosophy, German for technical subjects, etc.).

4. In theory there is no reason why the last word just entered on a page is not a revision rather than initial scribing: a word could have been chosen, passed to the motor apparatus for output, interrupted there by a higher process which was reviewing the text in parallel, and a new word substituted. An observer would have no way of knowing whether that word was the pristine "next" in an unreflected stream or the *n*th choice after *n-1* rejects.

5. In contrast, for programming, this stage is an important area for the development of specialized tools to help the program-modifiers learn about the foreign programs they have been asked to alter — since the modifier of a program is seldom the original author (cf. Miller 1978).

6. Consider the sentence "The plane stops in Dallas/Ft. Worth". Many word-processor users might well be suprised to learn that the fifth "word" (token) processed by the computer is Dallas/Ft.

7. In the IBM system, headings are indicated by the prefix letter "h", followed by an integer from 0 to 9; as the integer increases so does the fineness of subdivison. Thus, "h0" is for a major part of a document, "h1" for a chapter, "h2" for a major chapter subdivision, etc. The sequence of section numbers given in the text ("1.0 . . . 2.1211") could be produced by the following sequence of tags in the text: :h1. :h1. :h2. :h3. :h3. :h4. :h5.. The text associated with each level tag could be varied in font type, size, and boldness, according with the local formatter interpretation programs.

8. A very simple example is that of the word "past," for which the dictionary returns the following string of part of speech features: (past(ADJ) past(Noun Sing) past(Prep)). For other words, especially verbs, many other features and attributes are reported in addition to part-of-speech. The dictionary access method and structure are the responsibility of Roy Byrd, with assistance from Mary Neff, Martin Chodorow, and Barbara Kipfer (and, previously, from Barbara Snitzer and Lori Alperin).

9. Tone here refers to the "connotations of interpersonal attitudes" and is primarily a word-level phenomenon. While most dictionaries do not report these connotations (as a function of particular sense or context), there is certainly no reason that this type of information could not be added to on-line dictionaries.

10. Thus, one would want to flag all of the following for the same hackneyed aspect: "This council deems this action advisable," "Do you now deem that an advisable course?", "We have deemed this, and all other actions taken by the previous administration, as reprehensible and not advisable pursuant to the . . . " On the other hand one would probably not want to flag the following sentences in the same way: "We deem him honorable, whether or not his actions were advisable in the circumstances . . .," and "Most deemed it wise to go slow. We surely all believe in the most advisable courses of action." Detection of frazzle phrases under such variable circumstances requires the aid of a grammar analyzer in addition to the dictionary.

11. Grammar-checking in Russian, Latin, and Hungarian, for example, would be a very much richer field, given the large number of noun cases, declensions, and conjugations of these languages: German grammar-checking, while less complicated, still has four cases and three genders, thereby offering many opportunities for ungrammaticality — more than, say, Spanish.

12. This original EPISTLE grammar employed the NLP rule language and bottom-up parsing approach of George Heidorn (cf. Heidorn 1972). It has been implemented in LISP and, recently, in a varient of PL/I, and it could accurately be called a "syntactic grammar" in that almost no semantic information was used in obtaining reasonable parses for a wide coverage of English — something of a surprise to many of our colleagues. EPISTLE-1 was the first to demonstrate the feasbility of accomplishing quite sophisticated syntactic analysis and text-critiquing; its grammar was primarily the work of Karen Jensen. The parses of EPISTLE-1, while adequate for a variety of text critiques, were still "appropriate". For example, the structural descriptions were not always right, some grammatical roles were not identified (e.g., some indirect objects), and the syntactic association between discontinuous but related elements might not be recognized (e.g., in [14], the relation between "who" and "to"). In addition, while semantic features could

be added to the grammar rules, there were no general mechanisms to accomplish such things as sense selection, semantic-case assignments, knowledge-checking, quantification, and overall scoping.

13. The new EPISTLE-2 PROLOG version is designed to produce an "accurate" first parse ("rather than "approximate") to support a wider range of applications requiring more semantic interpretation — including document indexing, and mechanized translation. Semantic features and type hierarchies, grammatic slot-frames, and knowledge-checking are all employed to achieve correct sense-selection and case assignments integral to obtaining the "accurate" parse. A logical form can also be produced which represents the meaning of the sentence in predicate expression form. The semantic formalisms and PROLOG implementation are presently those of Michael McCord (cf. McCord, 1982, 1984, 1985), who also provided an extensive startup PROLOG grammar of English. With this beginning, Alexa McCray is developing a theoretically motivated system grammar, adapting McCord's interleaved syntactic/semantic approach to our requirements, ultimately to include handling of discourse-level phenomena.

14. Martin Chodorow is thanked for his insight into PROLOG and the grammar, and it appears that he has identified an elegant solution for the fragment and run-on problems.

15. The EPISTLE-1 grammar and critiquing function has been tested repeatedly on a set of 2K sentences and over a variety of texts, including business correspondence, student themes, and military manuals. Many sentences are given a reasonable parse, and most of the errors are detected — those that naturally occurred and those that were introduced specifically to test performance. On the other hand, a number of sentences received an incorrect parse, and the false-alarm rate (saying there's a grammar problem when the text is actually correct) is quite a bit higher than we wanted. Our plans for the EPISTLE-2 prototype are to create a testbed of 4-8K sentences (from perhaps a dozen genre types, mainly commercial), to develop automatic means for characterizing the features of a parse (comparing the latest parse of a sentence to that from the previous testing session, and to the "ideal" parse if available, and automatically classifying the nature of differences; this is the work of Mary Neff), and to classify the parse overall as perfect, acceptable for critiquing, or unacceptable, summary statistics will be provided for correctness (of the first, desired, parse), total number of parses, CPU time, and storage. Our goals are to develop an experimental system which provides at least acceptable parses for 98% of the sentences, detects 99% of the grammatical errors, has a false alarm error rate of less than 1%, and runs quickly and comfortably on an IBM PC-AT.

16. Yael Ravin has developed and implemented the style function.

17. A "fact" is either a proper name or a certain type of quantified noun phrase. Proper names are names of people, places, or things, and the latter two categories can get quite complicated. Quantified noun phrases include money amounts, physical dimensions, addresses, dates, phone numbers, etc.

18. References are explicit stylized pointers to other sections of the document and are of five types: (1) pointers to items in the reference bibliography (e.g., "McCord 1984"), (2) pointers to footnotes, (3) pointers to entities embedded in the text (e.g., formulae and examples), (4) pointers to non-textual entities (e.g., figures, tables, photos) and (5) pointers to structured text units (e.g., "see Appendix 1.2," "cf. Chapter 3;" a page-number reference may be considered such a pointer).

19. These are items other than straight text, to which reference is usually made in the text, such as footnotes, tables, formulae, etc. They also include other items to which no reference may be made but which are visually — and informationally — an important aspect of the document (e.g., company logo, printed telephone number, text border, dividing line between text and footnote, etc.) Repeating page elements, such as a running title or a security classification, are not considered as instances of this type.

20. In the summer of 1984 Joseph Janes and David Halpern of the School of Information Studies, Syracuse University, worked with the author to flesh out the detailed proposal for their respective fields of journalism and law. All details provided here are derived from that work program. The legal work was chosen for illustration (Halpern 1984a, b; Halpern and Miller 1984), but the journalistic work (Janes and Miller 1984) was as comprehensive and complete.

21. We find that students run words together, break words at other than hyphen points, leave out spaces between an end-of-sentence period and the first word of the next, capitalize in strange ways, omit obligatory punctuation, repeat words unintentionally, accidentally shift their hands on the keyboard while touch typing, introduce spurious characters (particularly when they hit a repeating key by mistake), skip lines randomly and do various other system-wrenching things, not even mentioning some of their peculiarities of syntactic structure.

REFERENCES

Byrd, R. J. "Word formation in natural language processing systems." *Proceedings of the 8th International Joint Conference on Artificial Intelligence,* 1983, 704-706 (Karlsruhe, Germany).

Byrd, R. J. "The ultimate dictionary users' guide," IBM Research Internal Report, 1984.

Byrd, R. J., and M. C. McCord "The lexical base for semantic interpretation in a prolog parser." Presented at the CUNY *Workshop on the Lexicon, Parsing, and Semantic Interpretation,* January 18, 1985, New York City.

Card, S. K., T. P. Moran, and A. Newell. "The manuscript editing task: A routine cognitive skill." XEROX Palo Alto Research Center Report No. P76-00082, 1976.

Carroll, J., J. Thomas, L. Miller, and H. Friedman. "Aspects of the solution structure in design problem solving." *American Journal of Psychology,* 1980, *93,* 269-284.

Cherry, L. L., M. L. Fox, L. T. Frase, P. S. Gingrich, S. A. Keenan, and N. H. Macdonald. "Computer aids for text analysis." *Bell Laboratories Record,* May/June, 1983, 10-16. 269-284.

Dalgish, G. M. "Microcomputers and teaching English as a second language." *City University of New York Research Monograph Series,* Report No. 7, September, 1984.

Durham, I., D. A. Lamb, and J. B. Saxe "Spelling correction in user interfaces." *Communications of the ACM,* 1983 *26,* 764-773.

Embley, D. W., and G. Nagy. "Behavioral aspects of text editors." *Computing Surveys,* 1981, *13,* 33-70.

Flower, L. *Problem-solving Strategies for Writing.* New York: Harcourt Brace Jovanovich, 1981.

Flower, L. and J. R. Hayes "The cognition of discovery: Defining a rhetorical problem." *College Composition and Communication,* 1980, *31,* 21-32.

Halpern, D. J. Unpublished taxonomy of legal writing genres, 1984a.

Halpern, D. J. Unpublished description of dwelling leases, 1984b.

Halpern D. J., and L. A. Miller. "Automating legal writing." Draft of a report on a classification of legal writing and the implementation of computer support for three selected sub-genres, August 27, 1984.

Heidorn, G. E., Jensen, K., Miller, L. A., Byrd, R. J., and Chodorow, M. S. "The EPISTLE text-critiquing system." *IBM Systems Journal,* 1982, *21,* 305-326.

IBM "Using the professional office system." IBM Publications Manual, Document Number SH20-5604-0, Chapter 12 ("Proof reading documents"), June, 1983.
IBM "Document composition facility: generalized markup language, starter set reference." IBM Publications Manual, Document Number SH20-9187-2, April, 1984.
Janes, J. W., and L. A. Miller. "Towards automated guiding and critiquing of journalistic writing." Draft of a report on work accomplished in classifying legal writing and implementing computer support of three sub-genres, August 9, 1984.
Jensen, K., and G. E. Heidorn "The fitted parse: 100% parsing capability in a syntactic grammar of English." *IBM Research Report,* RC 9729, 1982.
Jensen, K., G. E. Heidorn, L. A. Miller, and Y. Ravin "Parse fitting and prose fixing: Getting a hold on ill-formedness." *IBM Research Report,* RC10457, 1984.
Leerburger, B. A. "IBM spelling checker." *Think,* November/December 1981, 25-30.
Malhotra, A., Thomas, J., Carroll, J., and Miller, L. "Cognitive processes in design." *International Journal of Man-machine Studies,* 1980, *12,* 119-140.
Mathews, J. C., and D. Stevenson. *Designing Technical Reports: Writing for Audiences in Organizations.* Indianapolis: Bobbs-Merrill, 1976.
McCord, M. C. "Using slots and modifiers in logic grammars for natural language." *Artificial Intelligence, 18,* 327-367, 1982.
McCord, M. C. "Focalizers, the scoping problem, and semantic interpretation rules in logic grammars." To appear in *Logic Programming and its Applications,* D. Warren and M. van Caneghem, Eds., 1985.
McCord, M. C. "Semantic interpretation for the EPISTLE system." *Proc. Second International Logic Programming Conference,* 65-76, Uppsala, Sweden, 1984.
McCord, M. C. "Modular logic grammars." *IBM Research Report,* RC10936, 1985.
Miller, L. A. "Behavioral studies of the programming process." *IBM Research Report,* No. 7367, 1978.
Miller, L. A. "Project EPISTLE: A system for the automatic analysis of business correspondence." *Proceedings of The First Annual National Conference on Artificial Intelligence,* Stanford University, August, 1980, 280-282.
Miller, L. A. "Natural language text are not necessarily grammatical and unambiguous. Or even complete." *Proceedings of the Association for Computational Linguistics,* Toronto (June 18), 1982a. (Previously appeared as IBM Research Report RC 9441).
Miller, L. A. "How to write: a barely annotated bibliography." *IBM Research Report* RC 9687, November, 1982b.
Miller, L. A., and J. C. Thomas, Jr., "Behavioral issues in the use of interactive systems." *International Journal of Man-machine Studies,* 1977, *9,* 509-536.
Miller, L. A., G. E. Heidorn, and K. Jensen. "Text-critiquing with the EPISTLE system: An author's aid to better syntax." *AFIPS Proceedings of the National Computer Conference,* Chicago, May 4-7, 1981, 649-655.
Misek-Falkoff, L. D. "Computing, text markup, and 'speech acts': From outline to published book with authorial ease." Paper presented at the *Sixth International Conference on Computers and the Humanities,* North Carolina State University, Raleigh, June 8, 1983.
Morris, C. *Signs, Language, and Behavior.* New York: Prentice Hall, Inc., 1946.
Nancarrow, P. R., D. Ross, and L. Bridwell. "Word processors and the writing process: An annotated bibliography". Privately made available, English Department, University of Minnesota, 1982.
Orr, K. T. *Structured Systems Development.* New York: Yourdon Press, 1977.
Pollock, J. J., and A. Zamor. "Automatic spelling correction in scientific and scholarly text." *Communication of the ACM,* 1984, *27,* 358-368.

Roberts, T. L. "Evaluation of computer text editors."XEROX Palo Alto Research Center Report No. SSL 79-9, 1980.
Schwartz, H. J., and L. S. Bridwell. "A selected bibliography on computers in composition." *College Composition and Communication,* 1984, *35,* 71-76.
Seybold. "Computer aids for authors and editors: A natural extension of word processing and typesetting?" *The Seybold Report on Publishing Systems,* 1984, *13(10).*
Sekuler, R. "From quill to computer." *Psychology Today,* February 1985, 36-42.
Yourdon, E., and L. L. Constantine, *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design.* Englewood Cliffs, NJ: Prentice-Hall, Inc. 1979.