



Pergamon

**Computers
and
Composition**

Computers and Composition 19 (2002) 453–470

The politics of the program: MS WORD as the invisible grammarian[☆]

Tim McGee^{a,*}, Patricia Ericsson^b

^a Philadelphia University, School House Lane and Henry Avenue, Philadelphia, PA 19144, USA

^b Michigan Technological University, Houghton, MI 49931, USA

Abstract

Because of its widespread availability to writing students, the grammar and style checker in MS WORD deserves a thorough critique. Although recent scholarship has addressed general issues surrounding grammar and style checkers, we investigate this particular program in depth, focusing on its theoretical underpinnings. We contrast the approaches to grammar and style embodied in the software with those found in current composition pedagogy and conclude with suggestions that go beyond customizing the Grammar Checker to advocating more thorough discussions of the very notions of stylistic and grammatical correctness.

© 2002 Elsevier Science Inc. All rights reserved.

Keywords: Composition; Correctness; Current traditional rhetoric; Grammar; Language; Linguistics; Natural language processing; Pedagogy; Style

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.

Mark Weiser (1991)

1. Introduction

The idea for this article was conceived accidentally. As we exchanged drafts for a conference presentation, we were entertained by some of the inane recommendations that our word

[☆] Tim McGee and Patricia Ericsson are veteran techno-rhetoricians who have collaborated on several conference presentations and co-authored an article published in *Kairos: A Journal for Teachers of Writing in Webbed Environments* that is based on their experience as co-directors of the Dakota State University Online Writing Lab.

* Corresponding author.

Email addresses: mcgeet@phila.edu (T. McGee), plericss@mtu.edu (P. Ericsson).

processor's grammar and style checker suggested. Soon, casual entertainment led to questions: Who programs these checkers? What approach to language does this automated system represent? What might this ever-present corrective force be teaching students? Does a grammar and style checker support or undermine what composition teachers are trying to accomplish? As our list of questions grew longer, we realized that we had, serendipitously, found a topic for another conference presentation and for this article.

As our discussions progressed, [Cynthia L. Selfe and Richard Selfe's \(1994\)](#) enjoinder from "The Politics of the Interface" echoed regularly: Composition teachers need to be more than users of technology; they need to "think carefully about the implications of its use within their own classrooms" (p. 496). The Selfes encouraged us "to teach students and ourselves to recognize computer interfaces as noninnocent physical borders. . . cultural borders. . . and linguistic borders" (p. 495).

[Joel Haefner's \(1999\)](#) "Politics of the Code" recommended an initial focus. Haefner quoted Theodor Nelson's claim that "a computer language is a system for casting spell," and concluded his article by asserting that if Nelson is right, "then English instructors in computer-supported classrooms need to know something about the context and the necromancers of the code" (p. 338).

Because we have concerns about the uses of technology, particularly word processing software in the composition classroom, we focused on the style and grammar checking tool in MS WORD, the most widely used word processing software in the world. Its ubiquity is unquestionable. And its Grammar Checker (MSGC)¹ is now the product of an increasingly sophisticated branch of Microsoft's research efforts, the Natural Language Processing Group, whose particular linguistic approach exerts an undeniable, though largely invisible, influence upon writers. The specifics of that approach and the philosophies of language that it embodies begged some investigation.

We also became intrigued by how machine-checking of grammar and style alters the very concept of teaching grammar.² Its default status ("Check grammar as you type") makes grammar a primary concern by foregrounding correctness even while writers are in the drafting stage. It gives grammar and style (both narrowly defined by this digital program) a precedence, at least temporally, over content, mode of discourse, or other rhetorical concerns.³ This precedence could well have a negative effect on the development of writing ability. Although the field of composition and rhetoric enjoys some diversity of opinion as to what constitutes "best practice," there is considerable agreement about some classroom activities that do *not* improve student writing, first among which would have to be grammar instruction ([Braddock, Lloyd-Jones, & Schoer, 1963](#); [Hartwell, 1985](#)).⁴ The stronger indictments assert that formal grammar instruction is not just ineffective, but positively harmful if it consumes limited instructional time that could be dedicated to activities that actually help students become better writers. Where, however, does an artificially intelligent checker fit into that understanding of the place of grammar instruction in the composition curriculum? Does the checker provide instruction? Take up time? A related question is when, within a writing process, is the best time for a student to attend to those issues to which Grammar Checkers do attend? [Mike Rose \(1984\)](#) listed editing "too early in the composing process" as a primary cause of writer's block.⁵ And, significantly, among the five distinct meanings of grammar that Patrick Hartwell identified, to which does the MSGC attend?

Finally, we considered the strategies that users and teachers might employ to maximize the help (or minimize the harm) that the Grammar Checker in WORD could actually cause in (or around) a composition classroom. Various authors have recognized the opportunities for customizing the programs we use (Haefner, 1999; Haist, 1999; Selfe & Selfe, 1994). Caroline Haist concluded her thorough analysis of Grammar Checker in WORD 97 with a list of recommendations, the first of which is to “have students proofread their work to eliminate as many errors as possible before using the Grammar Checker,” a wise suggestion given her finding that “the more errors there are in a sentence, the more likely that Grammar Checker is to provide incorrect suggestions” (“Recommendations” section, 1999, par. 2). However, because the default setting in recent versions of WORD is “Check grammar as you type,” most users will do just the opposite of what Haist recommends. Furthermore, as Haefner noted, “customization assumes a single user for the software, not a lab or classroom environment where many students use the same software copy, the same workstation, many times a day” (p. 334). Finally, there is a limit to how much customizing any program will accept. As Fred Kemp (1992) wrote, “Computer software. . . no matter what flexibility it may claim or what ability to accept ‘user definition’ or modifying parameters, can never escape the instructional attitudes and even the ideology of its programmers and designers” (p. 9–10).

Our argument, then, is that the Grammar Checker in MS WORD represents an especially important piece of software for composition teachers to use critically for several reasons: its ubiquity, its near invisibility, its increasing power, its theoretical mismatch, and, in most cases, its actual conflict with and possible undermining of pedagogies that are now considered most effective for improving student writing.

2. Ubiquity of Microsoft’s Grammar Checker

In a purely numbers game, Microsoft wins the software war hands down. In the early 1990s, Microsoft claimed to have 35 million users of its operating system and software; however, since the monopoly controversy of the late 1990s, specific numbers are difficult, if not impossible, to find. Industry experts estimate Microsoft’s current share of the word processing software market at somewhere between 80 and 90%,⁶ and in 1997 Microsoft was aiming to become a major force in the educational software market, reversing their position earlier in that decade (Microsoft makes, 1997). The ever-expanding “Microsoft in Education” branch of its web site illustrates the corporation’s desire to promote its presence in the educational market. Simply by virtue of its ubiquity, Microsoft gets more “teachable moments” than the English teachers do. There are now just over 500,000 English teachers in the United States. Because WORD is on millions of desktops and Grammar Checker is turned on by default, it has many more, practically invisible, “over your shoulder”⁷ opportunities to be a grammar teacher than the typical English teacher—formerly one of the only purveyors of grammar instruction in a student’s writing experience.

3. Whose grammar is it?

This change in instructional personnel is no small matter. It reminded us of the Selfes’ warning that unless “those who are familiar with language and learning theory, who understand

issues raised by technology studies and cultural studies” pay close attention to the design of programs, “interface design will continue to be dominated primarily by computer scientists and will lack perspectives that could be contributed by humanist scholars” (1994, p. 498). However, the issue here goes deeper than interface design to become not only a matter of who is teaching grammar and how they are teaching it, but a matter of what they mean by “grammar” in the first place.

Some users may assert that the Grammar Checker in *WORD* is nothing but a mechanical delivery system for the grammar found in handbooks, and, therefore, represents no cause for alarm. Others, however, while accepting the view of MSGC as a mechanical handbook, would consider that to be an indictable offense, as handbook grammar is sometimes a dumb-downed version of “real” grammar,⁸ and simply tossing it at a struggling writer is far from effective pedagogy. We, however, contend that the content of MSGC is, in fact, quite different from a traditional handbook. And although it might be comforting if we could demonstrate that the grammar those computational linguists poured into MSGC represents an impoverished or debased form of what English teachers (wise humanists, they) have been dispensing for years, the reality is not so simple.

Yes, some MSGC content comes straight out of handbooks, and is therefore every bit as beneficial or useless as what can be found in some “school grammars.” And some points of careful usage that better grammar handbooks try to preserve get flattened by the binary Grammar Checker, thereby depriving users of a subtle marker of formal tone. For example, several recent handbooks insist upon “which” for nonrestrictive relative clauses and recommend “that” for restrictive clauses, but will allow “which” for restrictive clauses, noting that such usage indicates a more informal tone. The MSGC, however, takes a strictly binary approach. It flags any “which” that is not bracketed by commas, thereby suggesting that restrictive clauses ought to use a “that.” It is difficult to determine whether the MSGC does this because it’s easier for the program to handle a simple binary or because the computational linguists had loaded in the more restrictive prescriptions of Strunk and White, rather than the more accurate description of more recent handbooks.

Nevertheless, the Grammar Checker in *WORD* embodies a far more sophisticated approach to grammar than most users have noticed (sophisticated in terms of computational linguistics (CL), yet increasingly mongrelized in terms of grammars). Paradoxically, however, its increased sophistication may make it more dangerous than ever before. And although it may seem ironic if the wise humanists were suddenly to be trumped by those with less enlightened views of language, the history of English grammar would simply be repeating itself. Writing about the rise of prescriptive grammar in the eighteenth century, [Albert C. Baugh and Thomas Cable \(1993\)](#) reported that Joseph Priestley’s linguistic “tolerance, and good sense” lost out in the marketplace to Robert Lowth who was “much more conservative in his stand, a typical representative of the normative and prescriptive school of grammarians” (p. 269). About the flurry of grammars published in the late eighteenth century, Baugh and Cable remarked that “most of these books were the work of men with no special qualification for the thing they attempted to do” (p. 270). Two centuries later, a similar event may be occurring as those who are highly qualified to do CL start assuming functions that had traditionally been the bailiwick of English teachers.

4. Critical use

Heeding Haefner's call to examine "the context and necromancers of the code" behind the Grammar Checker requires looking into Microsoft's Research Group, first established in 1991. The establishment of these research labs indicated Microsoft's interest in "speech recognition, futuristic user interfaces and 3-D graphics-cutting-edge technologies" (Buderi, 1999, p. 45). The branch of Microsoft's Research Group that produced the MSGC is the Natural Language Processing Group, consisting of about 50 researchers—almost all of whom are computational linguists. Although some consider the field of CL synonymous with Natural Language Processing (NLP), the editors of *The Handbook of Natural Language Processing* assert "a clear demarcation between domain-specific theory on the one hand and practical development of computational language processing systems on the other." It is the latter, purely utilitarian approach that they put at the center of NLP, calling it the "least ambitious" of those fields that link computational ideas and human cognitive functions, limiting itself largely to producing "practical tools for the design of input–output modules" (Dale, Moisl, & Somers, 2000, pp. vi–vii). An epitome of such practicality would be a tool to check that aspect of written discourse that so many writers have some insecurity about—their grammar.

The Natural Language Processing Group hit paydirt when they produced a grammar and style checker better than the one that Microsoft had been licensing from Inso.⁹ When Microsoft installed its own checker in WORD 97, they not only improved their product and saved some money, but, according to Dan Ling (former Research Director at Microsoft and now the Vice-President of Microsoft Research), the sentence parsing program now built into every copy of WORD was "the foundation for building lots of different natural language tools" (qtd. in Buderi, 1999, p. 47). This foundation has much to do with Microsoft's long range goals regarding usability and office automation, but very little to do with the teaching of writing. George E. Heidorn, a researcher in Microsoft's NLP group, described the "product orientation" in the NLP research group as an "enormous benefit." Instead of being interested only in theoretical ideas and "interesting linguistic examples," researchers now pay attention to "real text as written by real people" (Heidorn, 2000, p. 205) due to the requirements of such a practical project. Their attention, however, is still that of linguists, and the linguist's interest in language is far different from that of the rhetorician and the composition teacher,¹⁰ and even farther still from that of the typical student writer. As Eric Johnson (1997) concluded, some "checkers have the ability to use sophisticated means to identify some grammatical blunders. . . but because they sometimes propose embarrassingly bad changes, the user must exercise great care in accepting suggestions" (par. 8). The "typical student writer" often has neither the grammatical expertise nor the patience to take the "great care" Johnson deemed necessary. In a similar vein, Caroline Haist found "most of the explanations Grammar Checker provides. . . helpful," but judged that "some are confusing and, in at least a couple of cases, just plain wrong" (1999, Introduction section, par. 6).

To become critical users of a program designed not by teachers of writing but by computational linguists, and made not for students of writing, but for workers in Microsoft's generic "Office," we must look closely at the approaches to language used by the NLP Group and then consider the points of contact and conflict with the approaches to language used by composition teachers.

5. Code upon code

Haefner warned that “how programmers go about writing code structures the way we (can) use computers” and Structured Programming’s “grounding in the American corporate environment [has] profound implications for the kinds of writing students do in computer-supported classrooms and the ways we teach writing” (1999, p. 337). The tenor of those implications are captured in Haefner’s earlier assertion that “the cultural imperative catalyzing SP has always been corporate productivity and profitability” (p. 330). Sociolinguist [Lesley Milroy \(1998\)](#) reminded us that English prescriptive grammar results from the fact that “by the eighteenth century Britain needed a standardized language to meet the needs of geographically scattered colonial government servants and to facilitate mass education” (p. 96) and that “since the goal of codification is to define a particular form as standard, this process entailed intolerance of a range of choices which speakers and writers had hitherto taken for granted” (p. 95). So, a computerized Grammar Checker, even one that was well designed with student writers in mind, would be likely to include two different kinds of codification constraining the choices of the novice writer.¹¹ And, as we will show, the Grammar Checker in WORD, because of its peculiar history, is anything but a program well designed for student writers.

6. More real than ideal

Building on Seymour Papert’s idea that computers are “naturally heuristic,” Fred Kemp claimed that we cannot “computerize any activity without having to (1) completely rethink the activity, including all the assumed behaviors that have become virtually invisible, and (2) discover in the new perspectives afforded by such rethinking, possibilities for actions that were never possible through the old perspectives” (qtd. in [Rickly, 2001](#), p. 10). Unfortunately, the NLP Group has not followed this “naturally heuristic” path in programming what users get in the MSGC. The real goal of the NLP Group is natural language processing: a user being able to employ natural language (actual speech) to interface with a computer. The implementation of the MSGC was just a happy coincidence for the Group.

To be responsible scholars and teachers, however, we must take note that no matter how theoretically sound and linguistically appropriate an approach the Group takes to natural language processing, its currently marketable byproduct (the MSGC) does not take a theoretically sound, rhetorically appropriate, pedagogically suitable approach to an ideal grammar and style checker. As [Eric Johnson \(1992\)](#) concluded:

The ideal grammar and style checker will be produced by a team consisting of linguists, computer programmers, and writing teachers. Linguists are obviously needed to develop the algorithms of word identification and sentence parsing. Programmers create the computer code to implement the algorithms. No less important are the writing teachers. They know the kind of advice that is likely to produce better writing. (par. 19)

A team composed of the members that Johnson suggested, aided by rhetoricians and compositionists, a team setting out to computerize an ideal grammar and style checker would, no doubt, aspire to what Papert and Kemp proposed: They would “rethink the activity, [investigate]

assumed behaviors. . . discover new perspectives,” and come up with an approach that truly would give them “a new lens through which to view the whole picture,” a lens that would show them “new ways of conceiving the very mission itself and supporting it” (qtd. in [Rickly, 2001](#), p. 10).

It is unfortunate for all users that the current MGSC is not “ideal” or even close to it. There is no “new lens,” no “new ways of conceiving” grammar or style in the MSGC. Although the programming might be complicated and the interface slick, users get recycled, often confusing advice about grammar and a mixed bag of style suggestions that don’t take into account current thinking on the grammar itself, good rhetorical theory, or pedagogical considerations. As we illustrate further, the problems of the MGSC are numerous, often dangerous, and largely invisible.

7. More power than intelligence

The Grammar Checker in MS WORD 97 and WORD 2000 (MSGC) is, indeed, a very different beast from the checkers found in previous versions of Microsoft’s word processing software. Although the casual user may have noticed little change, one reviewer claimed that WORD 97 was “the first word processor available that comes with a Grammar Checker worth using” ([Lowe, 1997](#), p. 36). The new checker was considerably more sophisticated than its predecessors, at least in terms of CL—a kind of sophistication, however, that may only exacerbate the problem caused by the software’s rhetorical naivete (a naivete that is understandable given that the Grammar Checker was not designed by composition teachers for student writers, and that the ultimate goal of its designers was not Standard Written English, but speech recognition).

According to Heidorn, “the NLP system that is behind the Microsoft GRAMMAR CHECKER [but not yet fully implemented] is a full-fledged natural language processing system that is also intended to be used for many other applications. It consists of a programming language and a runtime environment that are both specifically tailored to the needs of an NLP system” (2000, p. 182). One part of that processing system is described as a “bottom-up, multipath, chart-parsing algorithm [that] also makes use of both probabilistic information and a heuristic-scoring scheme to guide the process to a quicker and better termination point” (p. 184).

Although such power and sophistication have caused scholars to grant that the MSGC and other recent checkers demonstrate “undeniable gains in functionality over their predecessors,” critics still concluded that “their reliability and accuracy. . . have improved only minimally” ([Vernon, 2000](#), p. 331).

An example of increased functionality with questionable reliability can be seen in the Grammar Checker’s newfound ability to revise—not just flag—passive constructions. For example, when MSGC flags “Bill was left by Mary,” it suggests revising it to “Mary left Bill.” Unfortunately, however, when the current checker encounters the sentence “Bill was left by the side of the road,” it suggests “The side of the road left Bill.”

Due to gaffes like that,¹² many remain unimpressed with how the program handles language analysis, despite recognizing that the ability to revise passive sentences into active ones is quite a feat of artificial intelligence. Also, few users suspect that revising passive to active constructions

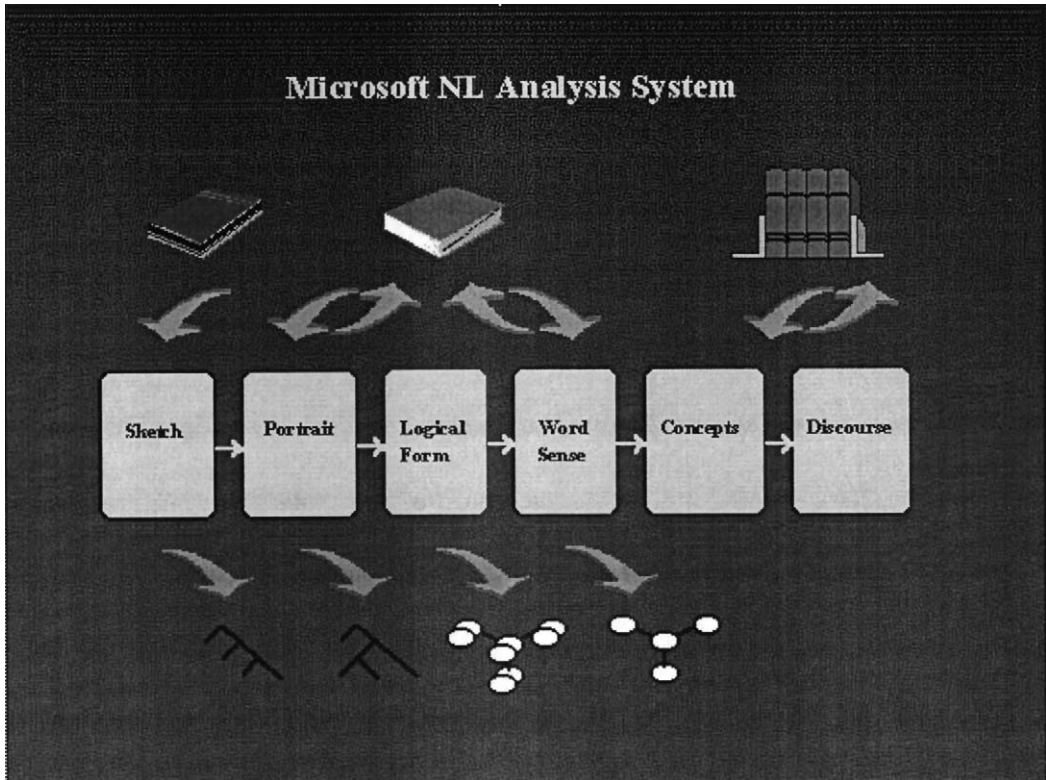


Fig. 1. Diagram of NLP analysis components. Source: Microsoft Research: Current Research at <<http://research.microsoft.com/nlp/analysis.asp>>.

is just the tip of an iceberg of potential functions hidden within Microsoft’s Natural Language Analysis System, only a fraction of which are enabled in the MSGC currently on the market. Figure 1 depicts all six of the stages that Microsoft’s full-fledged NL analysis system can perform upon text in their research lab.

At the top of the diagram three different kinds of texts are depicted: the input text to be analyzed, a dictionary (with input and output arrows occurring twice), and a shelf of books, the large knowledge base they call “MindNet” (also showing both input and output arrows). Below the texts are the six stages of analysis, representing the progression from lexical, through syntactic, then semantic, and culminating in discourse analysis.

The MSGC in currently available versions of WORD is limited largely to lexical and syntactic processing, whereas the NLP system behind the checker is well equipped to handle some kinds of semantic processing as well. Semantic processing occurs in the third stage of the Microsoft NLP analysis system, the syntactic portrait. In that stage, the NL analysis system attempts to “produce more reasonable attachments for some modifiers.” Heidorn explained how semantic reattachment “makes use of semantic relations that are produced automatically by analyzing the text of definitions and example sentences in an online dictionary. These semantic relations are stored in a rather large knowledge base that is now known as MindNet.” That is how

Microsoft NLP system knows “that it is more likely that a telescope is an instrument for seeing a bird, rather than a part of the bird in the sentence ‘I saw a bird with a telescope’” (p. 185).¹³

However, such a complex system cannot yet be packed in its entirety into the commercial word processor. “Fortunately,” according to Heidorn, “some applications of the Microsoft NLP system, such as the current Grammar Checker, can do an adequate job without doing semantic reattachment and, therefore, can be spared the added cost of that processing” (2000, p. 185)—adequate, perhaps, for the mature writer who wants help spotting the occasional slip in grammar or usage, but far from adequate for students with a tenuous grasp on Standard Written English.

If the system behind the MSGC is, in fact, so linguistically sophisticated, why would such a system mistake “the side of the road” for the unnamed agent who had abandoned Bill? This is partly due to gaps in the system’s collection of “factoids” and “multiword entries” (MWEs), bits of information that the currently installed system does possess about certain individual words or word groups. Thanks to a factoid, the very program that suggests revising “Bill was left by Mary” to “Mary left Bill” does not suggest that “Bill was left by April” be revised to “April left Bill.” Why? Because the software knows that “April” can represent a month. However, “by the side of” would not be a factoid but a multiword preposition stored as its own “MWE” in the dictionary. That is how MSGC knows to suggest “behind” as a replacement for “in back of.” Apparently, however, the system does not yet possess “by the side of” as an MWE for “beside,” because it never suggests that revision.

In addition to the increased powers of syntactic analysis, another improvement, first added in WORD 2000, is the software’s ability to flag successive nouns in groups of more than three. Listed under the “Style” options, this feature allows the user to guard against “Strings of several nouns that may be unclear, as in ‘The income tax office business practices remained the same’” (Microsoft, 2000). This feature is one of five new checking options that Alex Vernon said are “likely inspired by WORDPERFECT” (2000, p. 331). Whatever the inspiration, the added proscription against noun + noun + noun represents a new level of style (or grammar)¹⁴ checking of the sort addressed by Joseph Williams, who warns against the long compound noun phrase as a source of ambiguity (Williams, 2000, p. 91). This new addition to the MSGC could represent a slight tonic to its overall rhetorical naivete because linguistic expertise of the sort that Hartwell describes as “developed by psycholinguistic studies of comprehension” appears to be finding its way into the Grammar Checker. This is the sort of perspective that William J. Vande Kopple (1998) felt linguists could offer to compositionists because “research in linguistics and discourse analysis can provide powerful generalizations about how readers will respond to characteristics of sentences and texts,” generalizations about which “students deserve to know” (p. 6). So some of the added functionality in WORD 2000 appears to be getting rhetorically smart. Again, however, just how well suited this sort of instruction is for student writers is debatable. Vernon noted that “the target market for these products [is] business and the adult professional,” (p. 331) a remark similar to the response some instructors have had toward Joseph Williams’s *Style: Ten Lessons in Clarity and Grace*; namely, it represents a style manual particularly well suited to the mature writer of the impenetrable prose sometimes known as bureaucratese or sociologese, but it is a style manual not particularly well suited to the novice writer whose difficulties have more to do with invention and fluency than with turgidity. So much for the visible influences in the most recent version of the MSGC. We now turn to some of its unseen influences.

8. Disappearing powers of MSGC

Two ways that the MSGC exerts its influence are the corporate cache of Microsoft itself and the interface of WORD—powers that are all the more influential as they have become unnoticed norms. Although tarnished by the monopoly trials that began in the late 1990s, the corporate image of Microsoft still shines as the standard bearer of the operating system and software world. Industry watchers keep a close eye on Microsoft's every move, its stock serving as a bellwether for the entire technology sector. Its brand name alone gives the Microsoft Grammar Checker a kind of influence that is difficult (perhaps impossible) to calculate.

WORD's interface is another visible factor with almost invisible power, as its polished surface has become so familiar that it has disappeared below the threshold of our notice. Granted, occasional new additions catch some attention, not all of which is positive. (Why the animated paper clip seems to arouse such antipathy in some users is hard to say; is it the simple intrusiveness or the bizarre coquetry of its eye movements?) In any event, the overall slickness of WORD's interface is something now so familiar we no longer notice it as an incredibly polished product constantly persuading us of its quality and dependability.

Other aspects of the program's influence are encoded into its default settings. Because so many users remain unaware of the options available in the Grammar Checker, millions have grown as accustomed to the squiggly green lines as they had become to the red ones signaling spelling errors. Grammar and style are constantly being checked in real time except for those users savvy enough to have turned off that option. And even the savvy user gets grammar and style advice automatically when choosing to spellcheck a document.

Some might argue that the green squiggles do render the checker visible, especially for those accustomed to word processing before the arrival of WORD 97. Many users may simply learn to ignore the green line or, more accurately, think that they have learned to ignore the green line. However, not acting on the line's recommendation or not bothering to inquire into its meaning is not the same as being uninfluenced by it. Some users have reported switching "which" to "that" whenever a "which" is flagged, but learning to ignore the full-sentence underlines (as they usually indicate "long-sentence," and those users don't mind writing long ones).

Although some users may have doubts about artificial intelligence after seeing the Spielberg/Kubrick film, *AI*, the intelligence built into the MSGC isn't nearly as ominous as that in the movie, but does lend another invisible force to the program. The invisible intelligence of a supposedly smart machine has a definite appeal. Fred Kemp commented that to many, the "computer. . . projects the universal ethos of science itself" (1992, p. 10). When this ethos and a trust in artificial intelligence are combined with the supposition that the smart people at Microsoft know more about grammar and style than the typical user, the Grammar Checker gains considerable power. More significant, however, may be the issue of consistency. Unlike teachers who may occasionally contradict themselves and who frequently contradict each other (making "Can I use 'I'?" one of the most frequently asked questions in every composition class), MSGC is as consistent as only a machine can be. In every situation, the Grammar Checker will respond identically to the same grammar and style issues. Students, moreso than the general population, have a near desperate need for certainties and "right" answers; a computer program gives them those certainties more readily than all-too-human English teachers.

Of particular interest to us and to others who are interested in the politics of the interface, code, and software is the entire politics of standardized language. This issue is invisible to most users, especially to student writers. Given that their school-based language instruction is largely prescriptive and that few, if any, precollege students have been let in on the secret that the prescriptions of Standard English are neither natural, nor logical, nor productive of language that is “better” (in any meaningful sense of the word) than nonstandard Englishes, most simply accept the authority of this smart machine to police their grammar, just as they have heretofore accepted the authority of their smart teachers. Although many might smart from this power relationship, for most, it has operated invisibly, as just one of the many givens of the school environment. For them, the situation simply could not be otherwise. It is just what is. “Good grammar” is a fact of school life, and the MSGC is licensed to check their grammar by the same invisible authority that licenses the rest of the adult world they are being socialized into.

Although all previous issues are vitally important to us as composition teachers and scholars, perhaps none is as significant as the possible misfit the MSGC might have with our objectives as teachers of writing. Not surprisingly, our conclusions on that question are as discouraging as the majority of our findings.

9. The MSGC contradicts good pedagogy

Throughout the recent history of composition (as well as the entire history of rhetoric), various approaches to teaching writing have existed simultaneously. No one theory or pedagogy has ever taken hold as “gospel.” This variety, according to [James Berlin \(1996\)](#), tends to proliferate “in proportion to the freedom tolerated in the society involved” (p. 138). Despite this multiplicity, we can look at current composition theory and pedagogy to get a sense of what is now considered best practice—an important notion to consider if we are going to judge whether the MSGC might support or undermine those attempting to enact such practice.

Best practice is characterized by a concern for the social nature of writing and an eye to the total ecology in which writing is incubated, produced, and consumed. Current best practice acknowledges that writing—whether it be invention, process, production or consumption—is a multilayered, collaborative, often digitally enacted, postmodern ecology. [Marilyn Cooper \(1989\)](#), one of the first scholars to posit an ecological model of writing, explained that this kind of a system “postulates dynamic interlocking systems that structure the social activity of writing” (p. 7). These ecological systems are continually in flux, steadily being created and recreated by readers, writers, and teachers in a complex web of interacting possibilities that those involved encounter. No matter which subtle flavor it takes on, best practice in the new millennium is decidedly social.

The nearly polar opposite of a social/ecological approach to teaching composition is the pedagogy commonly known as *current traditional rhetoric*. This method of teaching is characterized by an overwhelming concern with forms and obsessive attention to precise, correct language. Although scholars as notable as [Robert Connors \(1997\)](#) objected to the term *current traditional* (pp. 4–7), [Debra Hawhee \(1999\)](#) claimed that current traditional rhetoric forms “a discipline in both senses of the word” because current traditional rhetoric has the ingredients of a discipline: “a subject, a ‘body of knowledge,’ a body of precepts for students to learn

and follow”; in addition, it disciplines students by “constructing them as aberrant individuals” (p. 521). The discipline of current traditional rhetoric is passed down from teacher to student and then is replicated in the students who become the next generation of teachers. The precepts of current traditional rhetoric are encoded in handbooks and workbooks that were considered ineffective for the teaching of writing as early as the 1880s (Connors, 1997, p. 117).

Contemporary critics of Grammar Checkers have commented that “style and grammar packages are generally based on an overly narrow—and erroneous—vision of ‘correct’ language use” (Selfe & Selfe, 1994, p. 489) or have claimed that simply tossing a handbook at a struggling writer represents some pretty bad pedagogy. Kemp posited that “computers can process text in only the most superficial of senses; computers cannot grasp the meaning in the text” (1992, p. 14).¹⁵ The least effective, empirically dismissed approach to the improvement of writing is the approach presented by the MSGC. It is concerned primarily with prescriptive issues of usage and surface concerns of style. Even in its screen appearance, it harkens back to the red pencil of the obsessive English teacher who bled over “mistakes” and paid little or no attention to the quality of thinking. The Grammar Checker in WORD 97, according to one reviewer, “attempts to be a full-fledged writing coach” (Campbell, 1997, p. 109), but the bulk of its comments are based on a digital, right/wrong, binary paradigm that make writing seem to be exactly what it is not: “Writing is not a mistake to be corrected, something broken to be fixed, a gap to be filled, or a wrong to be righted” (Kemp, 1992, p. 23). Sadly, MSGC is primarily a current traditional machine—a machine that looks dangerously smart, especially to users insecure about grammar and usage.

Even though materials on the Microsoft web site tout WORD 2000 as having “new features [that] also make it an excellent tool for collaboration” (Word 2000, 2001, par. 1) and explain how WORD can be used to enhance the writing process, making WORD a useful teaching tool requires that both students and teachers overlook the most obvious critique that WORD makes—red and green highlighted commentary on correctness. In addition, the ability to change styles may seem like a computer-smart guarantee that the writer is going to “get it right” for her particular audience simply by setting a level of formality that the computer verifies. This surface consideration of language that is audience appropriate is almost laughably simplistic.

10. Disarming the Microsoft necromancer

At the end of our efforts to “pay attention” to this particular software, we are tempted to encourage all writing teachers to throw away their copies of MS WORD and to ban it from their schools. However, if we actually had the power to instigate such a revolution, the most prevalent grammar and style checker on our computer desktops would already be different. Despite this inability, we believe composition teachers are not powerless, even in the face of the Microsoft giant; we believe we can play David to this Goliath. Slaying the giant probably isn’t possible (and perhaps now that we know so much more about the software, it might not even be desirable), so how can composition teachers minimize the harms it could cause?

Carolyn Haist provided several suggestions to minimize the harm of the MSGC at the end of her 1999 article, and others have provided similar suggestions (Spinks, Wells, & Meche, 1997; Vernon, 2000). In giving our presentation on this topic at the Computers and Writing

Conference in 2000, we still felt quite confident in providing a list of things that a composition teacher could do to ameliorate problems MSGC might present. And we still believe that a list of steps can help a composition teacher—to some degree. However, because of what our critique has uncovered, we find using practical, how-to, or step-by-step guides too insignificant—rather like just throwing rocks at Goliath instead of aiming a slingshot at him.

What can we provide as a metaphorical slingshot? What can we employ to make our aim more accurate and our impacts more substantive? We believe the answer lies in knowledge. To begin with, leaving decisions about grammar up to Microsoft is simply unacceptable. All English teachers need more than a basic understanding of prescriptive grammar; we need a rich understanding, informed by scholarship like that of [Patrick Hartwell \(1985\)](#) and others who have considered grammar in a more complex way. We need to understand that “grammar” can be used politically to limit access to position, power, and wealth. We need to understand the subtleties of grammar far better than most of us do.

We need to spend time digging around in software before using it. When a new grammar-and-style checker appears, we need to take time to dig into it, rummage around in its “options”¹⁶ and “defaults” to see what kind of a beast it is. (Most of us need to do this immediately with the checkers that are currently on our desktops and in our labs—even if they aren’t new.) We need to be confident enough with the technology to “play” with it, to open it up for inspection, and to think about what we see and what it means for us and for students in our classes. Mindlessly accepting a piece of software is irresponsible—even if everyone in the world is using it, even if we can’t really change it, even if we’re afraid of breaking it. We are in complete accord with what Selfe told us in her 1998 CCC address: “*As composition teachers, deciding whether or not to use technology in our classes is simply not the point—we have to pay attention to technology*” (Selfe, 1999, p. 415, original emphasis).

In this same address, Selfe also told us that we need to get students involved in critiques of technology:

Composition teachers, language arts teachers, and other literacy specialists need to recognize that the relevance of technology in the English studies disciplines is not simply a matter of helping student work effectively with communication software and hardware, but, rather, also a matter of helping them to understand and be able to assess—to pay attention to—the social, economic, and pedagogical implications of new communication technologies and technological initiatives that affect their lives. (p. 432)

This might appear to be difficult, but we believe that it doesn’t have to be. If teachers are confident that the software can’t really be broken (and it can’t—Microsoft has made sure of that), then we can feel free to ask students to help us investigate it. Whatever age students we teach, many know more about playing with computers than we do, and they are usually more eager to do it. We can ask them to look at the options that the MSGC offers. Once they are inside the software, we can encourage questions about the various grammar, usage, and style choices that Microsoft offers. We think that more than a few students will notice that under the “Formal” option in WORD 97, the checker has all but three possibilities turned on—and one possibility that is *not on* is to check for gender-specific words. If we ask them to check into the explanations of the grammar and style options, they can check to see if their notions of “Commonly confused words” match up with the ones Microsoft provides.¹⁷

The opportunities that this kind of investigation can provide us are practically unlimited. It gives us a chance not only to talk about grammar in context, but also to think about and talk with students about the politics of Microsoft, how the checker was constructed, and what it might mean for different kinds of students from differing backgrounds.

Finally, those of us in positions to influence teacher education programs must insist that the courses in these programs include up-to-date composition theory, broad-based and complex instruction in grammar, investigations of the ideological underpinnings of technology, and enough hands-on work with computers and software that teachers are eager to go below the primary interface. Those of us in positions to influence hiring practices need to make sure that the people we hire can interrogate technology.

In the end what we need is quite revolutionary, even though the revolution might not include slaying Goliath. Our revolution must go to the heart of how we think about and interact with technology; this revolution is about knowledge, education, and action. We need a revolt in the ranks—people currently teaching with these technologies need to take a critical interest. And we need to include our students in this uprising.

But none of this is really new—those in-the-know have been calling for these kinds of changes for at least a decade. What is new, we believe, is our call for these changes following a thorough critique of a piece of software that can powerfully affect our teaching. We are confident that our critique, one that uncovers a largely invisible influence in our teaching lives, will spur some to action. We hope that what we have uncovered makes the invisible grammarian of MSGC less mysterious, less able to cast spells. We hope we have wrested some power from the Microsoft necromancers and put it back into more capable hands—yours.

Notes

1. Although its full and most accurate name might be Microsoft's Grammar and Style Checker, for the sake of simplicity we will refer to the tool that appears in MS WORD 97 and WORD 2000 as the Grammar Checker or MSGC for short.
2. In addition to altering the concept of teaching grammar, machine-checking of grammar could, to some extent, alter the concept of writing itself. As [Anne Herrington and Charles Moran \(2001\)](#) concluded their critique of the machines now being used to score student essays, "Writing *to* the machine. . . creates what is for now an unprecedented and unnatural rhetorical situation. . . . Writing to the machine desensitizes us as writers" (pp. 496–97).
3. Proponents of the Grammar Checker might argue that because users can first choose one of five "styles," the Grammar Checker actually gives priority to the notions of audience and occasion; however, many users remain unaware that there are five styles from which to choose or that each of those can be customized—few of the participants in two workshops we have conducted even knew about these options.
4. Hartwell admitted that the grammar issue was still controversial in 1985 and listed papers defending the teaching of formal grammar as well as those attacking it. A 1995 *Composition Chronicle* article with the title "Grammar Making a Comeback in Composition Teaching" ([McCleary, 1995](#)), attests to grammar's perseverance. The controversy continues to this day, but, thanks in part to Hartwell's work, the complex notion of grammar

has been unpacked to the point of including five distinctly different meanings, a few of which we will employ later in this article.

5. In his attempt to identify the “cognitive variables involved in writer’s block,” Rose (1984) listed six reasons that some writers block, four of which could well be exacerbated by using MSGC in its default mode. In addition to editing too early, three other causes are these: “(1) the rules by which they guide their composing processes are rigid, inappropriately invoked, or incorrect. . . ; (5) they invoke conflicting rules, assumptions, plans, and strategies; and (6) they evaluate their writing with inappropriate criteria or criteria that are inadequately understood” (p. 4).
6. Estimates vary widely depending on how the data are sampled; for example, Microsoft’s dominance appears smaller when law offices figure heavily in the sample as WORD PERFECT still commands a large share of the legal word processing market, holding 82% of that market as recently as 1998 (Phelps, 1998, par. 2.).
7. An online *Product Enhancements Guide* for WORD once proudly asserted that WORD can “act as the reader over your shoulder as you type.” That document, previously available at <<http://www.microsoft.com/TechNet/Word/prodfact/wd97peg.asp>>, has been removed.
8. Hartwell (1985), expanding on “The Three Meanings of Grammar” that W. Nelson Francis offered in 1954, provided five distinct meanings, including grammar 1, “The Grammar in Our Heads,” and grammar 2, “The Scientific Grammar of Linguists.” The third, which Francis referred to as “Linguistic Etiquette,” according to Hartwell, “is, of course, not grammar at all, but usage” (p. 110). Grammar 4 is school grammar, a user friendly but slightly inaccurate version of scientific grammar that teachers use with students, is found in most handbooks. Hartwell described the “rules” of “common school grammars” as “inadequate to the facts of written language” (1985, p. 119).
9. That product had first been available for PCs in 1989 as CORRECTTEXT GRAMMAR CORRECTION SYSTEM from Houghton–Mifflin, later from Inso Corporation. No slouch of an application, it was one of two commercial systems identified by Robert Dale (1996) as “the first products to use anything related to the parsing technologies developed in the research field.” He went on to predict, accurately, that “as machines become more powerful, and as broad-coverage grammars become more feasible, we can expect to see more of the CPU-hungry techniques developed in research labs finding their way into products” (section 7.5.1, par. 4).
10. The gulf between linguists and compositionists is perhaps greatest around the grammar issue. Discussing what he calls the “romantic” position that “stylistic grammars. . . have little place in the teaching of composition,” Hartwell (1985) asserted that “this position rests on a theory of language ultimately philosophical rather than linguistic” and suggested that we “witness, for example, the contempt for linguists in Ann Berthoff’s *The Making of Meaning: Metaphors, Models, and Maxims for Writing Teachers*” (p. 124). William J. Vande Kopple (1998) argued that “In the last twenty or fifty years, research on language has gone from an area that specialists in composition and rhetoric took quite seriously to one that specialists now pay little attention to” (p. 4).
11. Some will argue that constraint, especially in the area of standard English and preferred usage, is exactly what novice writers need; however, most compositionists will agree that such constraint is best applied late in the composing process, not “as you type.”

12. The Grammar Checker also duly flags “Me and my friends went to the store,” but suggests that the writer should have written “My friends and me went to the store.”
13. This “rather large knowledge base” represents the solution to the persistent shortcoming of the artificial intelligence use of computers that Kemp (1992) explained as follows: “Despite more than 30 years of effort, artificial intelligence has foundered on the problem of implanting the kinds of experience into computers which provides a linguistic context capable of handling the extraordinary contextualism of natural language” (p. 16).
14. Hartwell’s grammar 5 is “Stylistic Grammar,” which he says enjoys “two fully articulated positions” that he labels the “romantic” and the “classic,” offering Joseph Williams’s *Style: Ten Lessons in Clarity and Grace* as an example of the latter position.
15. Interestingly, Kemp (1992) attributed the ultimate failure of “artificial intelligence use of the computer” to the fact that “the problem lay in the computer’s inability to employ natural language” (p. 13). The MSGC represents the best commercial effort, so far, to address that part of the problem, but, as its gaffes indicate, its ability to employ (and analyze) natural language is still quite limited.
16. Getting into the innards of the MGSC is as easy as clicking on the “Options” tab when the grammar and spelling checker is running. From there, clicking on “Settings” will open a range of possible choices. To see even more of the workings, clicking on “Help” and typing “grammar and style options” in the search field will give the user a moderately informative explanation of each possible grammar and style option.
17. We are still amused and amazed that “augur/auger” appears in this list of commonly confused words but that “affect/effect” does not. (Perhaps the computational linguists developed their list in collaboration with a school of Mining and Divination.) We assume students will also experience some incredulity.

Tim McGee directs the Instructional Technology Program at Philadelphia University. His scholarly interests include computer-based writing pedagogy, the rhetoric of science, and argument from the locus of quantity. He has published in *Text Technology*, *JTW: The Journal of Teaching Writing* and the section on oral presentations for the *Allyn and Bacon Guide to Writing*.

Patricia Ericsson is a PhD candidate in rhetoric and technical communication at Michigan Technological University. Her scholarly work focuses on agency in technological environments, composition theory, rhetoric, and assessment. She has published in *Text Technology* and *ACE Journal*, and she has also contributed chapters to various books.

References

- Baugh, Albert C., & Cable, Thomas. (1993). *A history of the English language* (4th ed.). Englewood Cliffs: Prentice Hall.

- Berlin, James. (1996). *Rhetorics, poetics and cultures: Refiguring college English studies*. Urbana, IL: National Council of Teachers of English.
- Braddock, Richard, Lloyd-Jones, Richard, & Schoer, Lowell. (1963). *Research in written composition*. Urbana, IL: National Council of Teachers of English.
- Buderi, Robert. (1999). Software's ultimate sandbox. *Technology Review*, 102(1), 44–51.
- Campbell, George. (1997). WORD 97. *PC World*, 15(2), 109–110.
- Connors, Robert. (1997). *Composition-rhetoric: Backgrounds, theory, pedagogy*. Pittsburgh: University of Pittsburgh Press.
- Cooper, Marilyn. (1989). The ecology of writing. In Marilyn Cooper & Michael Holzman (Eds.), *Writing as social action* (pp. 1–13). Portsmouth, NJ: Boynton/Cook.
- Dale, Robert. (1996). Computer assistance in text creation and editing. In Ronald A. Cole, Joseph Mariani, Hans Uszkoreit, Annie Zaenen, & Victor Zue (Eds.), *Survey of the state of the art in human language technology*. Retrieved March 21, 2001, from <<http://cslu.cse.ogi.edu/HLTsurvey/ch7node7.html#SECTION75>>.
- Dale, Robert, Moist, Herman, & Somers, H. L. (2000). *Handbook of natural language processing*. New York: Marcel Dekker.
- Haefner, Joel. (1999). The politics of the code. *Computers and Composition*, 16, 325–339.
- Haist, Caroline. (1999, November). An evaluation of Microsoft WORD 97's Grammar Checker. *Syntax in the schools*, 16(2). Retrieved July 1, 2001, from <<http://www2.pct.edu/courses/evavra/ATEG/Mono/Haist/GCHECK.htm>>.
- Hartwell, Patrick. (1985). Grammar, grammars, and the teaching of grammar. *College English*, 47, 105–127.
- Hawhee, Debra. (1999). Composition history and the Harbrace College Handbook. *College Composition and Communication*, 50(3), 504–523.
- Heidorn, George E. (2000). Intelligent writing assistance. In Robert Dale, Hermann Moisl, & Harold Sommers (Eds.), *Handbook of natural language processing* (pp. 181–207). New York: Marcel Dekker.
- Herrington, Anne, & Moran, Charles. (2001). What happens when machines read our students writing? *College English*, 63, 480–499.
- Johnson, Eric. (1992). The ideal grammar and style checker. *Text Technology*, 2(4). Retrieved July 1, 2001, from <<http://www.dsu.edu/~johnsone/ideal.html>>.
- Johnson, Eric. (1997). The current state of grammar and style checkers. *Text Technology*, 7(1). Retrieved July 1, 2001, from <<http://www.dsu.edu/~johnsone/grammar.html>>.
- Kemp, Fred. (1992). Who programmed this? Examining the instructional attitudes of writing-support software. *Computers and Composition*, 10, 9–24.
- Lowe, Doug. (1997). *More WORD 97 for WINDOWS for dummies*. Foster City, CA: IDG Books.
- McCleary, Bill. (1995). Grammar making a comeback in composition teaching. *Composition Chronicle: Newsletter for Writing Teachers*, 8(6), 1–4.
- Microsoft. (2000). Microsoft WORD 2000 [Computer Software] online help files. Redmond, WA: Microsoft Corporation.
- Microsoft makes move into educational software. (1997, September 24). *Lubbock-Avalanche Journal*. Retrieved July 1, 2001, from <<http://lubbockonline.com/news/092597/microsoft.htm>>.
- Milroy, Lesley. (1998). Myth 12: Bad grammar is slovenly. In Laurie Bauer & Peter Trudgill (Eds.), *Language myths*. London: Penguin Books.
- Phelps, Alan. (1998). Why not Word? Competitors still challenge Microsoft's word processor dominance. In *Smart computing in plain English*. Retrieved July 12, 2001, from <<http://www.smartcomputing.com/editorial/article>>.
- Rickly, Rebecca. (2001, May). Technology, institutional assessment, and big brother. *Council of Writing Program Administrators President's Newsletter*, 8–10.
- Rose, Mike. (1984). *Writer's block: The cognitive dimension*. Carbondale, IL: Southern Illinois University Press.
- Selfe, Cynthia L. (1999). Technology and literacy: A story about the perils of not paying attention. *College Composition and Communication*, 50(3), 411–436.
- Selfe, Cynthia L., & Selfe, Richard, Jr. (1994). The politics of the interface: Power and its exercise in electronic contact zones. *College Composition and Communication*, 45(4), 480–504.
- Spinks, Nelda H., Wells, Barron W., & Meche, Melanie. (1997, October). Customizing grammar checking. *Business Education Forum*, 24–26.

- Vande Kopple, William J. (1998, April 1–4). Current research on language and its status with composition teachers. Paper presented at the annual meeting of the conference on college composition and communication (ERIC ED418408).
- Vernon, Alex. (2000). Computerized Grammar Checkers 2000: Capabilities, limitations, and pedagogical possibilities. *Computers and Composition*, 17, 329–350.
- Weiser, Mark. (1991, September). The computer for the twenty-first century. *Scientific American*, 94–104.
- Williams, Joseph. (2000). *Style: Ten lessons in clarity and grace* (6th ed.). New York: Longman.
- Word 2000. (2001). Microsoft in education. Retrieved July 1, 2001, from <<http://www.microsoft.com/education/tutorial/classroom/o2k/word.asp>>.