Computerised writing aids: do they really help?

Carel Jansen

At present, more and more software is being launched which is intended to assist authors in solving all kinds of writing problems. However, most studies into the effects of using these new writing aids show disappointing results, especially when grammar and style checkers are concerned. The main problem with these programs is that they lack the knowledge needed to analyse sentence and text structures in an appropriate way.
It seems to be more promising to develop advisory databases that authors – when confronted with a writing problem – can easily access. CATS is a prototype of such an advisory database. CATS definitely is not a panacea for writing problems, but as the results of a first, small-scale study indicate, software like CATS could eventually grow into a really useful tool for professional writers.

Technical communicators nowadays seem to hold a threefold relation with information technology. They write *about* information technology, producing for instance computer manuals, instructions for consumer electronic products or company safety rules. They also write *for* modern information technology, when they create on-line help texts or adapt information for a hypertext or multi-media system. And finally, technical communicators write *by means of* products of information technology. At present, it is hard to imagine a technical communicator writing without using an advanced word processor including such features as a spelling checker, automatic hyphenation and a thesaurus.[1] In many contributions to this volume, a call is made for attention to the problems that technical communicators encounter when writing about or for technology. In this contribution, however, the focus will be on the third aspect of the relation between communicators and information technology: the application of modern writing tools.

1 Grammar and style checkers

According to sales figures, especially in the English-speaking world, *grammar and style checkers* (GSCs) are growing into a more and more popular tool for profes-

sional (and non-professional) writers. Vendors suggest that GSCs can identify all kinds of grammatical errors in a text, make reliable judgments about the style of the document, calculate a number of readability indexes and suggest important corrections and improvements. Thus, GSCs are often claimed to add an important extra value to word processing software.[2]

The *Writer's Workbench*, developed by AT&T Bell, can be regarded as the forefather of GSCs (cf. Van der Geest et al., 1987). This software package, running under UNIX, consists of 28 programs, among which *Organisation*, which prints out the first and last sentence of each paragraph, *Findbe*, which underlines all forms of to be, and *Diction*, which flags any of 525 wordy, overused, misused, sexist or inflated words and expressions.

Grammatik, *Rightwriter* and *Correct Grammar* are the names of some of the most popular GSCs. They belong to the new generation of writer's aids that run on personal computers and that can easily interact with word processors such as *Word* and *WordPerfect*. Just like the *Writer's Workbench*, these new GSCs rely heavily on simple pattern-matching and counting. Grammatical rules are included only at the morphological level; at the syntactic and semantic levels, the parsing abilities of GSCs are still very limited. The reason for that seems to be quite obvious. Just as in the field of automatic translation, the linguistic problems that arise when automatic sentence analyses (or even more difficult, text analyses) are to be achieved, are not overcome by far (cf. Dijkstra et al., 1990, and Nijholt, 1992).

An exception seems to be IBM's *Critique*, a grammar and style checker running under UNIX, which makes use of a parser generator called PNLP. In Van den Akker (1992), where the development of a Dutch version of *Critique* is described, a clear insight is offered into the tremendous efforts that have to be made, as well as in the problems that are to be overcome, when trying to construct a GSC based on serious syntax-based sentence analyses. It proves that in order to perform its duties, *Dutch Critique* needs some 350 different lexical attributes, assigned to over 40,000 lemmata, and referred to in several hundreds of grammar rules. Van den Akker convincingly demonstrates that given the fact that virtually any sentence is ambiguous, the construction of an adequate algorithm which ranks alternative parses by order of their relative probability, is extremely important – and difficult.

How difficult it is is shown in an evaluation study performed by Dijkstra et al. (1990). They compared *Critique* (the English version) to three less complex GSCs. It proved that *Critique* hardly performed any better than the others did.[3] Dijkstra et al. state:

> [. . .] though Critique's grammar has been under development for some years and is able to make a correct and complete list of some very complicated sentences, it is still not perfect. It misses some basic verb cluster, double comparatives, punctuation, and style errors [. . .] it shows some surprising flaws due perhaps to its complex architecture.

2 Studies on grammar and style checkers

Baring the problems with *Critique* in mind, it is not hard to understand that developers of most GSCs, at least for the time being, decided not to make use of advanced parsing facilities but to base their products on much more restricted forms of pattern-matching. That does not necessarily mean that these GSCs are not useful to the writer. If used with care, perhaps they could be a valuable, though not perfect, aid for professional authors in trying to locate writing errors and stylistic shortcomings, or for teachers in trying to draw their students' attention to text features that could be improved.

However, research does not support this supposition. So far, not many studies have been undertaken into the effect of using GSCs. The studies that have been carried out mostly did not take place in a professional setting but in an educational setting and the results reported are not very encouraging. To give an example, Day (1988) asked a group of college students who had become familiar with the *Writer's Workbench* if using this software had led them to any changes in their writing process. Out of the 67 respondents who answered this question, 58 said 'no' or 'not really', and only 8 students answered 'to some degree'. Of course, asking questions like this, can only be considered as an indirect way to find out if GSCs do have an impact on the writing product or the writing process. It is possible, although not very likely, that there have been real changes in the product or the process that the students involved have not been aware of.

This methodological problem is circumvented in a study by Peek et al. (1988). They compared the texts written by two groups of students who had taken a writing course. The experimental group had received feedback from the GSC *Rightwriter*; the control group had received human feedback on their papers. After the course, a panel of four judges rated the texts of both the experimental and the control group. Analysis of the ratings for experimental and control groups revealed no statistically significant differences.

A problem with this study is that the 'treatment' for the experimental group was very limited. The students only received feedback from *Rightwriter* on one assignment. Besides, this feedback was combined with feedback from their teacher, so that it is hard to make a real comparison between the effect of feedback from *Rightwriter* on the one hand and the effect of human feedback on the other.

The general picture emerging from experiments like these is that – methodologically weak as they are – they certainly do not support the claim that writers learn from using GSCs. But perhaps the conclusions from other types of evaluation studies, aimed more directly at analysing the outcomes that the programs present to the users, can shed a different light on the value of GSCs.

An example of such an evaluation study was performed by Van Mansom (1993). He evaluated the outcomes of a text analysis by *Correct Grammar*, *Rightwriter 5*

and *Grammatik 5*. As test material he used three different texts, written in a professional setting (approximately 20 pages in total). He presented the texts to the GSCs, and also to an expert on editing English-written texts. In comparing the outcomes, he first determined the real problems, the grammar and style problems that the human expert had indicated, and the software problems, the problems indicated by the various GSCs. Then three new categories were identified:

- *hits* problems indicated by the program and belonging to the category of real problems;
- *misses* problems belonging to the category of real problems and *not* indicated by the program;
- *false alarms* problems indicated by the program but *not* belonging to the category of real problems.

Van Mansom's results are summarised in table 1.

	hits (percentage out of real problems)	**misses** (percentage out of real problems)	**false alarms** (percentage out of signalled problems)
Grammatik	21.4	78.6	84.3
Rightwriter	14.2	85.8	84.1
Correct Grammar	12.2	87.8	84.4

Table 1: Hits, misses, and false alarms in GSC analyses

Obviously, the three GSCs evaluated in this study have only very limited help to offer in discovering real writing problems. No more than one out of five textual shortcomings was identified, four out of five were missed. Perhaps even more disturbing was the overall high percentage of false alarms: more than 80% of the messages was best ignored by the user of the software.

Van Mansom's study is not unique. Dijkstra et al. (1990), for instance, evaluated earlier versions of Critique, Correct Grammar, Grammatik and Rightwriter. The authors report *hit* percentages of 47.6%, 50.0%, 41.2% and 33.7% respectively. The *miss* percentages accordingly amounted to 52.4%, 50.0%, 58.8% and 66.3%. The percentages of *false alarms* for Critique, Correct Grammar, Grammatik and Rightwriter were 7.3%, 27.7%, 53.6% and 29.2% respectively.

At first sight, this evaluation seems to turn out considerably more positive for the GSCs than the study undertaken by Van Mansom. It should be noted, however, that of the nine texts that Dijkstra et al. used to test the GSCs, six were 'demo texts' supplied with one of the GSCs that were evaluated. When constructing this kind of text the vendors understandably tend to include grammatical and stylistic problems which can be identified by their own (and often other) GSCs and to exclude

problems which GSCs cannot handle. Using these demo texts in an evaluation study may easily lead to overestimating the capabilities of the software under consideration.

A more recent study, this time into French GSCs, was undertaken by Vanneste (1993), who reports the results of an evaluation of Hugo Plus, Grammatik (French) and GramR. In testing these GSCs, Vanneste found 16.9%, 29.7% and 13.0% *hits* and 83.1%, 70,3% and 87.0 % *misses* respectively. How many *false alarms*, as defined by Van Mansom, were found cannot be inferred directly from the data Vanneste presents.

Neither these effect studies nor the evaluation studies discussed above give rise to much optimism about the usefulness of the present GSCs for professional writers. Given the linguistic problems that have to be overcome before adequate text representations can be yielded, great improvements in the performance of GSCs are not to be expected. Perhaps, then, it would be better to shift attention to another type of writing aid: on-line advisory databases.

3 Advisory systems

If it turns out to be so hard to develop software that automatically offers adequate writing advice, then – at least for the time being – it may be more effective to try and develop *on-line advisory systems* that inform the author *on request*. Clearly, such systems have the disadvantage that they only offer solutions to problems that writers have experienced themselves and that nothing is done about textual shortcomings that are outside the author's field of vision. An important advantage, on the other hand, is that the writer is not left with a wrong idea about the qualities and weak spots in his text, as may very well be the case when a GSC is used.

Examples of on-line advisory systems for English writers are *Editorial Advisor* and the software versions of the *Shipley Style Guide* and Strunk & White's *Elements of Style*. An advisory system for Dutch has been under development in Utrecht since 1991. The system is called CATS: an acronym for Computer Adviezen voor TekstSchrijvers (Computer Advice for Text Writers).[4] A number of issues involving CATS will be discussed in the remainder of this article.

4 CATS: a Dutch on-line advisory system under development

On-line advisory systems can only be expected to be effective if they have clear advantages compared to their hard copy counterparts. As long as writers can find all the information they need just as quickly in a more conventional medium, there

is a fair chance that the on-line tool will hardly be used. An important question which then arises in the development of advisory software is which demands the product has to meet to be more useful to the writer than the traditional text book.

The starting point for the development team were the three tasks that CATS users have to perform in order to benefit from the advice the program has to offer.[5]

1 The user has to look for the appropriate advice; hence CATS should be *easily accessible*.
2 The user has to decide if the advice found is really useful; hence CATS should allow easy toggle switching to and from the word processor.
3 The user has to integrate the advice in the text: when appropriate the writer should be able to make use of the standard phrases or examples offered by the advisory text; hence CATS should provide convenient copying facilities.

To meet these three demands, the following features were incorporated into the system. Firstly, CATS presents the user with different ways to get access to the right advice. After having started the word processor (which can be any commercially available program as long as it runs under MS DOS), and after having activated the memory-resident CATS software by pressing the key combination CTRL-ALT-C, users can either:

- move through a series of menus which lead from general categories, such as advice on press reports or letters of application, to more specific advice, for instance on how to formulate a headline or on how to structure a résumé; or
- use an alphabetical index to search for advice on the specific writing problem they are facing; or
- start by looking at one piece of advice and then decide to move to another one; to get there they can click on one of the *hot spots* in the first recommendation: words that are highlighted to indicate that they function as hypertext nodes which link one CATS topic to another.[6]

There is still one other possibility for users to reach the piece of advice they are looking for: CATS permits *context-sensitive* searching. If users know an adequate expression for the matter they want to be advised on, they can type this expression in their ordinary word-processor screen, and activate CATS only then by pressing CTRL-ALT-C. If the word or phrase that was typed in is part of the list of search expressions included in CATS, users are automatically presented with the advisory text they were looking for. If not, CATS's main menu appears, and the users can either start a search action from that point on, or return from CATS to their word processor by pressing [F7].

To allow toggle switching to and from the word processor, CATS advice is presented in a window which only partially overlaps the normal word-processing screen that the user is looking at. The place where the CATS window appears depends on the location of the cursor in the word-processor screen. If the cursor is

in the upper half, the advice appears in the lower half and vice versa. This split-screen presentation allows users to look at their own text and the CATS text at hand at the same time, thus reducing memory problems in applying the advice.

To allow copying a CATS recommendation fully or partially into the text at hand, users are offered a procedural option: mark a block in the recommendation they are looking at and press [F4]. The block then is integrated in their own text, starting from the location of the cursor. This facility prevents users from having to retype what was already presented to them, while not obliging them to use the literal wording of the CATS recommendation in their own product. After having copied the recommendation into their text, they can edit it at will.

Easy searching, switching and copying facilities were not the only demands kept in mind in developing CATS. High quality content is at least as important if an advisory system is to be effective. Hence, the CATS developers have tried make the advisory texts meet the following demands.

- Reliability: the advice should be correct and complete.
- Relevance: the advice should be applicable to the specific communication situation for which the writer has to produce his material.
- Surprise: at least part of the advice should not be completely predictable for the user.

To meet these requirements, the advisory texts included in the CATS prototypes developed so far (cf. note 4), have been based on what the most prominent text books have to say on the subjects that the prototypes deal with. Furthermore, technical provisions have been made so that users themselves can modify the content of the CATS version they want to work with. This allows individuals and organisations to adapt the advisory system to their specific needs by adding or skipping any advice they like or by changing the hyperlink structure of their original CATS version.

An important question, of course, is whether the efforts made to make CATS meet the demands that the developers regard as important have resulted in an advisory system that qualifies as a really helpful computerised writing aid. The only way that such a question can be answered is by performing empirical studies into the effects of the software tool involved. A number of such studies are being planned or carried out at this moment.

However, only one small scale study has been completed to date. Schneider & Tanis (1993) presented 8 subjects with a CATS version on how to write a letter of application. The subjects were asked to bring a personnel advertisement of their choice to the search room and to write a letter to match this advertisement, using the word processor and consulting CATS whenever they felt that might be useful. They were also asked to think aloud as much as possible and were observed while performing their task. Afterwards they were interviewed on their opinions about various aspects of the advisory system they had been working with.

It emerged that all of the subjects frequently consulted CATS and used many of the advisory texts presented to them, especially where the content and the structure of a letter of application were concerned. Somewhat less frequently used were the advisory texts on matters of style and layout. All in all, the advice offered was evaluated positively; none of the subjects felt that CATS in general included too little or superfluous advice. All subjects felt that having used CATS had helped them in writing a better letter than they would normally have written.[7]

That the subjects appreciated CATS as a whole does not mean that there was no criticism at all. For instance, negative remarks were made on the sometimes unclear nodes indicating a hyperlink to another item of advice. In such cases, the subjects were somewhat unpleasantly surprised that after having clicked a certain node, they were presented with advice on another matter than the one they had hoped to find information on. Another kind of criticism concerned the technical problems that some of the subjects experienced when trying to copy all or part of a recommendation into their own letters. It turned out that the way this facility (highly appreciated in itself) could be used was far from self-evident, and that the on-line technical help that this CATS version offered did not always suffice to solve the problem.

The results of this study, small-scale as it was, indicates that advisory systems like CATS can have a future. Important conditions for success, however, seem to be a high quality content as well as optimal access and application facilities, based on elaborate studies into the effects of various ways of presenting the advice to the user. Only when these conditions are met can disappointment about the usefulness of new writing tools be prevented.

Notes

1 A special type of word processor which might prove to be of interest to technical writers in the near future is what are termed SDPs: *Structured Document Processors*. SDPs can be regarded as controlled writing environments that prompt writers for information (somewhat like form filling), and automatically format document design features. For an overview, see Norman & Grider (1992). They make clear that SDPs are best used for specialised, formulaic writing. No studies are reported into the effects of using SDPs on the writing process or on the text quality.

2 In the *Grammatik 5 User's Guide*, for instance, the introduction states, 'Grammatik carries grammar analysis to the limits of computational technology' and 'Whether you are a corporate manager, a government official, a professor, or a student, Grammatik will help you communicate more effectively' (p.1).

3 The exact results are discussed in the next section.

4 CATS is a co-production of *NICEWARE* Utrecht and Utrecht University (Centre for Language and Communication). CATS is written in PDC-PROLOG, runs under MS DOS and is memory-resident. Prototypes have been developed containing advice on such issues as how to write a readable text, how to write an adequate press report and how to design an effective access structure for software manuals. CATS prototypes running under MS-WINDOWS will be forthcoming.

5 See also De Vet (1993)

6 For a discussion of hypertext and research literature considering it, see Van der Geest (1994).

7 In this study, the opinions of the subjects could not be checked against real quality differences in the letters written with or without using CATS. In future experiments, for which designs including pre- and post-tests and experimental vs. control groups are being planned, more objective ways are foreseen of establishing the effects of using CATS on writing products. (Cf. Jansen, 1993)

References

Akker, D. van den (1992), Language technology at IBM Nederland. In W. ter Stal et al., *Twente workshop on language technology 2; Linguistic engineering: tools and products*. Enschede, p.93-101. Memoranda Informatica p.92-29, University of Twente.

Day, J.T. (1988), Writer's workbench: a useful aid, but not a cure-all. *Computers and composition*, 6 (1), p.63-77.

Dijkstra, A., Huls, C., Claassen, W., Cremers, L., & Van Vliembergen, E. (1990), *Grammar checkers checked*. Tilburg: Stichting Mens-Machine Communicatie en Taaltechnologie, University of Brabant. MMC Research Report no. 9.

Geest, Th. van der (1994), Hypertext: writing and reading in a non-linear medium. (this volume).

Geest, Th. van der, Jansen, C., & Looijmans, P. (1987), Schrijven en computers: de stand van zaken [Writing and computers: state of the art]. *Tijdschrift voor taalbeheersing*, 9, p.113-133.

Jansen, C. (1993), *CATS in gebruik: op weg naar een on-line adviesprogramma voor tekstschrijvers [CATS in use: towards an on-line advisory program for writers]*. Paper presented at the VIOT congress in Ghent (December 1993).

Mansom, C. van (1993), *Onderzoek naar de bruikbaarheid van moderne style & grammar checkers voor de schrijver [Study into the usefulness of modern style and grammar checkers]* Graduation thesis. Utrecht University.

Norman, R., & Grider, D. (1992), Structured document processors: implications for technical writing. *Technical communication quarterly*, 1, p.5-21.

Nijholt, A. (1992), Linguistic engineering: a survey. In W. ter Stal et al., *Twente workshop on language technology 2; Linguistic engineering: tools and products*. Enschede: University of Twente. Memoranda Informatica 92-29, p.1-22.

Peek, G.S, Eubanks, T., May, C., & Heil, P. (1988), The efficacy of syntax checkers on the quality of accounting students' writing. *Computers and Composition*, 6, p.81-109.

Schneider, M. & Tanis, P. (1993) *Solliciteren met CATS. Afsluitend werkstuk onderzoekscollege over schrijfadvies-software [Writing a letter of application using CATS. Concluding item of a graduate class on advisory software for writing]*. Utrecht University.

Vanneste, A. (1993), *Checking grammar checkers*, Paper read at the Second Conference on Functional Communication Quality, Antwerp, June 1993.

Vet, D. de (1993), *CATS: a real help for professional writers?* Paper read at the Second Conference on Functional Communication Quality, Antwerp, June 1993.

About the author

Carel Jansen graduated in Dutch Literature and Linguistics at Utrecht University, where he was also awarded his doctorate in 1989. The thesis dealt with communication problems between government and citizens. From 1976 to 1989 Carel Jansen worked as an assistant professor at the University of Twente. In 1989 he returned to Utrecht where he became an associate professor. Since September 1993; he also holds a chair of technical communication at Eindhoven University of Technology. Most of his research is on the subject of instructional texts. In close collaboration with his colleague Michaël Steehouder, he has published articles and books on topics such as communication problems, forms design and (technical) writing.