

Assessment of the minimalist approach to computer user documentation

Hans van der Meij and Ard W. Lazonder

The minimalist approach (Carroll, 1990a) advocates the development of a radically different type of manual when compared to a conventional one. For example, the manual should proceed almost directly to procedural skills development rather than building a conceptual model first. It ought to focus on authentic tasks practised in context, as opposed to mock exercises and isolated practice. In addition, it should stimulate users to exploit their knowledge and thinking, as opposed to imposing the writer's view and discussing everything that users should see or know.

In the first part of the paper the construction of a tutorial based on the minimalist principles is described. A parallel is drawn with constructivism with which minimalism shares important notions of instruction. In the second part, an experiment is described in which the minimal manual was tested against a conventional one. The outcome favoured the new manual. For example, minimal manual users completed about 50% more tasks successfully on a performance test and displayed significantly more self-reliance (e.g. more self-initiated error-recoveries, and fewer manual consultations).

Keywords: human-computer interaction, documentation, user manuals

Carroll *et al.* (1987) have claimed that a minimal manual helps first-time users learn how to operate a word-processor in 40% less time and with 58% better retention of skills than a conventional manual. Surely, if you were about to develop a manual, wouldn't these claims make you wonder whether minimalism is the answer for you?

Before you start devising your own minimal manual, however, it is useful to note that several critics have argued that the design principles have not been operationalized in enough detail for practitioners to construct their own minimal manual (Hallgren, 1992; Horn, 1992; Nickerson, 1991; Tripp, 1990). In addition, criticisms have also been expressed with regard to its empirical

Department of Instructional Technology, Twente University, PO Box 217, 7500 AE Enschede, the Netherlands, E-mail: meij@edte.utwente.nl

claims. Probably one of the most important arguments here has been the problem that the minimal manual was 'based' on the control manual. Hence, it might just have been a better second effort (Nickerson, 1991).

This study begins with an overview of the minimalist design principles, before presenting an empirical study investigating the minimalist claims.

Minimalist principles

According to Carroll, a minimal manual is based on four main principles (see Carroll, 1990a; 1990b; 1993; Lazonder and Van der Meij, 1992; *in press*):

- action-oriented nature,
- text-optimization,
- support of error recognition and recovery,
- modularity.

The various design principles that are subsumed under these aspects are summarized in Table 1.

The main characteristics of minimalism are described in the following sections. Some of these principles are illustrated with examples from the minimal manual (MM) used in the study. Corresponding examples from the conventional (i.e. control) manual (CM) are also discussed.

Some of the principles of Table 1 are general, rather than exclusive to minimalism, and have already found their way into conventional manuals. Consequently, the MM does *not* differ from the CM on all of the above principles. For example, both manuals use relatively short and simple sentences, signal action information and concentrate on a menu-oriented means of access to the program. Sometimes the difference is also more a matter of degree rather than of presence versus absence. The examples plus Figures 1 and 2 should clarify these points and give a good impression of the two manuals used in the experiment.

Action-orientation (AO)

The MM focuses on task execution, on functionality for the user. Its primary goal is to help the novice accomplish basic tasks. The MM therefore emphasises the procedural part of a program, leaving out all information that does not directly relate to it. The CM, in contrast, gives considerable attention to the declarative aspect of the program — to building a conceptual model. In addition, it also gives slightly more procedural information.

The two manuals have minor and major differences on this point. The most important ones concern the AO-principles 2, 4, 5 and 6 of Table 1. The latter two principles will be used for illustration.

Principle 5: Capitalize on exploration and problem solving

This principle has frequently been interpreted as a plea for leaving out basic action information. This is a mistake, however. As Figure 1 shows, all the necessary action steps for 'Searching a text', a basic task, are described in the MM.

Table 1. Design principles for a minimal manual

Action-orientation (AO)

- 1 Concentrate on one main means of access to the program (e.g. using menus versus using a combination of menus and function keys).
- 2 Introduce alternative methods sparingly and show their advantages 'in context'.
- 3 Select safe tasks and sequence tasks safely.
- 4 Focus on real tasks.
- 5 Capitalize on exploration and problem-solving (e.g. by creating 'On your Own' sections).
- 6 Give users the opportunity to act early on.
- 7 Signal action information.

Optimal use of text (OT)

- 1 Replace unnecessary jargon.
- 2 Give action-oriented explanations close to where the knowledge is to be used.
- 3 Do not spell out everything.
- 4 Write headings that accommodate to the users' prior knowledge and their need for information.
- 5 Write in short, simple sentences.
- 6 Use words and sentences that invite users to become actively involved (e.g. use commands and questions instead of declaratives).

Support error recognition and recovery (ER)

- 1 Error information should support detection, diagnosis and recovery.
- 2 Give error information when errors are costly, time-consuming or when they are liable to occur frequently.
- 3 Error information should lead to fast recovery, but always treat general recovery methods before specific ones.
- 4 Give 'on-the-spot' error information.
- 5 Use a standard formula for error information.
- 6 Signal error information.

Modularity (M)

- 1 Make chapters short.
 - 2 Provide closure of chapters.
 - 3 No explicit referencing between chapters.
-

Instead, the MM encourages exploration in sections such as 'On your own' in Figure 1. In the illustrated section, users are invited to explore other options that are also available for 'Searching a text'. It is in sections such as these that the action steps are left out so that users must rely on working things out for themselves. Such problem-solving activities may enhance skills transfer. Note also that the section explicitly invites users to act: "Try them and see" (principle: OT6).

Information with which users can explore is also presented in the CM. For example, Figure 2 shows a section that may prompt users to explore (i.e. the section: "If you want to . . . search function."). However, this information is

Searching a Text

You can position the cursor quickly to a word or part of a sentence by searching for this text.

1. Position the cursor at the beginning of the file
2. Go to the menubar and press twice on the → key
3. Choose the command FORWARD and press the ENTER key

WordPerfect asks what you want to search for. Check to see if the prompt → **Search:** is on your screen.

4. Type any word(s) from the text
5. Press the F2 key

If the code [Hrt] appears after the word, you have pressed the ENTER key instead of the F2 key. Remove the code by pressing the BACKSPACE key. Then press F2 for searching the word(s).

WordPerfect automatically positions the cursor at the first occurrence of the word(s) in the text.

*If you hear a bleep and the text *Not found* appears, you may have made a typing mistake. Try again.*

On your own: Searching text

The commands NEXT and PREVIOUS enable you to find out if the word you have been searching can also be found elsewhere in the text. You can find these commands under the SEARCH option. Try them and see.

- Action-oriented heading (OT4)
- Implicit, general goal statement (OT3)
- Act early on (AO6)
- Use a menu-oriented approach (AO1)
- Signal action information (AO7)
- Implicit screen prompt (OT3)
- Incomplete linkage information for reference (OT3)
- Use words/sentences that stimulate actions (OT6)
- All error information principles (ER1 to ER6)
- All error information principles (ER1 to ER6)
- Capitalize on problem solving (AO5)
- Use words/sentences that stimulate actions (OT6)

Figure 1. The left column presents an illustrative page of the minimal manual. The right column characterises the corresponding design principles. (Abbreviations + numbers, e.g. AO5, refer to the design principles of Table 1)

presented more as 'nice-to-know' information than as an invitation for action. The text 'discusses' the option. In addition, there is no distinct signalling (e.g. a separate place or typography) to alert users to the fact that they may want to try it out.

Principle 6: Give users the opportunity to act early on

This is one of the most distinct characteristics of the MM. For example, the MM introduces working with a word-processor on page one and then presents the conventions used in the manual on another half a page before users are prompted to start their computer. In contrast, the CM has six introductory pages, with more information about word processing and some information about keyboard usage.

4.2 How to Search a Text

In editing a file it may often be necessary to search for a text. Words or parts of a sentence may, for example, be 'somewhere behind the screen' and the fastest way to position the cursor there is by searching. Searching for a text can be done in two ways: forward and backwards.

4.2.1 Directing a Forward Search

To search forward from the starting point of the cursor (towards the end of the file) you do as follows:

1. Press [ALT] and choose the SEARCH option
2. Press [I] to select the command FORWARD
3. Press [ENTER]

On the statusbar you should now see the prompt
-> Search:

4. Type the text you want to search for; this text will appear on the statusbar after the prompt
5. Press [F2] to start the search

If you want to continue the forward search from this last cursor position to search for another occurrence of the text, you should select the command NEXT. This command can also be found under the SEARCH option. So, with this command you renew the search function.

Please Note (1): Do not press [ENTER] to end the text you want searched. This will lead to the code [Hrt] (of 'Hard return') behind it, meaning that the program will search for your text including a hard return. In short: this will be found only when your text appears at the end of a sentence or paragraph.

Figure 2. The left column presents an illustrative page of the conventional manual. The right column characterises the design principles that were shared and not-shared with minimalism. (Codes such as 'not-AO5' refer to related information but not a shared minimalist design principle of Table 1)

- Action-oriented heading (OT4)
- Advance organizer (not-AO6)
- Explicit, specific goal statement (not-OT3)
- Redundancy (not-OT3)
- Use a menu-oriented approach (AO1)
- Signal action information (AO7)
- Full and explicit linkage information (not-OT3)
- Explicit linkage information (not-OT3)
- No error information (not-ER1+6)
- Declarative words/sentences (not-AO5+OT6)
- No distinct signal for error information (not-ER6)
- Explanation of system message (not-OT3)
- Declarative words/sentences (not-OT6)
- Redundancy (not-OT3)

Figure 1 shows how this principle is applied in other sections of the MM. As you can see the heading 'Searching a text' is followed by a short description of the corresponding goal statement. In contrast, the CM begins with a discussion about search possibilities in general, before introducing a specific goal (see Figure 2).

Optimal use of text (OT)

The text in the MM serves to help users act. It is, in a sense, 'written around' the skills that users should develop, leaving out any information that is not directly relevant for task execution. Carroll *et al.* (1987) speak of 'slashing the verbiage'.

The process of adapting the text to the users' actions is more subtle than weeding out the excess, however. For example, explanations are not simply reduced, they are presented immediately before or after the relevant action steps. Moreover, they describe what something 'does' rather than what it 'is'. For this reason the principle is referred to as 'optimal use of text'.

There are major differences between the two manuals for the principles 2, 3 and 6 that relate to optimal use of text (see Table 1). These, and principle 5, are illustrated below.

Principle 2: Give brief, action-oriented explanations close to where the knowledge is to be used

Explanations are coupled to the tasks in the MM. So, where the CM presents the keyboard and states what certain keys can accomplish in the introduction, the MM introduces these keys 'on-the-job', when users need them to achieve a goal.

Also, as discussed earlier, unlike the CM, the MM does *not* use prior descriptions of generalities (see Figures 1 and 2).

Principle 3: Do not spell out everything

The MM leaves out information that can easily be inferred. Clearly it is difficult to judge when such a situation applies. For this reason, and because this principle is among the most important and innovative ideas of minimalism, it will be illustrated in some detail.

Figure 1 presents a subtle example with regard to the goal statement for 'Searching a text'. As you can see the text does *not* mention the fact that users will be carrying out a forward search. Instead, this should become clear from the way in which the action steps are described and sequenced:

- (1) start at the beginning of the file (this makes a forward search the only option);
- (2) ... and 'press twice on the → key' (by not being explicit here, users are stimulated to study the screen and to notice they are about to select the option 'search');
- (3) ... select 'forward' (the first explicit cue of *forward* searching).

Another example of this principle relates to the linkage information in the manuals. Linkage information tries to co-ordinate the processing of manual, screen and keyboard. There are two ways in which the MM stimulates these co-ordinating actions. First, the MM prompts users to study the screen to find a message. Second, the MM describes the message that appears on the screen (i.e. when users need explicit information, for example as a point of reference), but does not detail its place. Most frequently this will be given in the form of commands or questions about the message. For example, "Check if the following sentence appears on the screen: "Name Document...", and "...->Search:..." (see Figure 1). In contrast, the CM always fully informs users of what they should extract from the screen, and often the location is added: "On the statusbar you should now see the prompt: ->Search:". This information is frequently presented as a declarative sentence.

A third illustration of this principle concerns the action steps. The action steps in the MM are always 'bare-bone' descriptions indicating *what* to do. There are never any comments; it is simply "Type . . .", "Press . . ." and so on. In contrast, some of the action steps in the CM are more detailed. For example, they may also describe what will happen, as in: "Type the text you want to search for; this text will appear on the statusbar after the prompt." (see Figure 2).

Principle 5: Write in short, simple sentences

Clearly this principle was applied to both manuals. Word counts revealed a mean sentence length in words of 10.7 and 14.5 for the MM and CM respectively. Both measures suggest that the manuals require less than a fifth-grade reading level.

As a check on sentence quality, a linguist was asked to correct the (Dutch) texts of the two manuals where necessary (see method section).

Support error recognition and recovery

Carroll and his colleagues (1984, 1987) have been among the first to recognize that users should develop skills both for doing things (i.e. constructive skills) and for undoing things that have gone wrong (i.e. corrective skills) (cf. Allwood, 1984). Being able to detect, diagnose and correct errors is an important skill for all users who need to master a computer program. The provision of error-information is therefore a key feature of the MM.

With regard to this main principle, the differences between the MM and the CM are enormous. Like most conventional manuals, the CM presents little error information. Correction keys such as backspace, delete and F1 are discussed, of course, but on the whole, the error information in the CM is given infrequently and does not receive special attention. One can thus say that the MM and CM differ strongly on nearly all aspects of support for error recognition and recovery. Principles 4 and 5 are illustrated.

Principle 4: Give 'on-the-spot' error information

When making a mistake, early detection is probably vital for recovery. If too long a period has passed between the error and its detection, backtracking the actions and correction may become almost impossible. Minimalism therefore stresses the need for 'on-the-spot' error information.

In the MM, nearly every method is accompanied by error information. As Figure 1 shows, there is error information after step 5. This action is error-prone because the users of the word processor (should) by then have become accustomed to ending their actions by pressing 'Enter'. Here, this means they will make a mistake. From the users' point of view the mistake is logical and it is likely to be made frequently. Pilot studies substantiated this and led to the inclusion of the error information.

There is also error information after the statement concerning the result of the user's actions. Here the information merely supports the error detection that is already visibly and audibly signalled by the program. It conveys the diagnostic and recovery information that users may need.

Note that there is *no* error information after the first three actions. This is due to the fact that the linkage information already supports detection and users can correct their actions easily with a general recovery method (e.g. by pressing F1).

Principle 5: Use a standard formula for error information

Error information should support the detection, diagnosis and recovery of mistakes in a fixed sequence. Detection should come first, then diagnosis and then recovery (Lazonder, *in press*). As users who have *not* made a mistake may also read the error information, it seems important to present the detection information as a proviso. In combination with what users might do next, this leads to the standard "If . . . (detection) then . . . (cause) then . . . (correction)" formula.

This is illustrated in the two blocks of error information in Figure 1, which is signalled by being written in italics. For example, in the first block, detection information is presented by the sentence: "If the code . . . word". The diagnostic information follows after the comma: "you have . . . F2 key". And the correction information comes after that: "Remove the code . . . word(s)".

The CM gives some error information in the section starting with "Please note (1)" (see Figure 2). In this section there is information for detection and diagnosis, but not for correction. Hence, users may not know how to proceed after making this mistake. In addition, the information is presented more as an aside (hence the "please note"), than as a distinct error cue. It looks like an afterthought and integration with the user's actions (and the on-the-spot characteristic of error information in the MM) is missing.

Modularity

One of the main reasons for using a modular approach in documentation is to accommodate browsing (see Arnold, 1988). With chapters that are self-contained, users who wish to read parts of the manual at will can do so. Another important reason is that the modularity enables users to stop at convenient points. The two manuals differ more or less on all three principles of modularity, but only the first principle will be illustrated.

Principle 1: Make chapters short

The differences for the first principle can easily be illustrated by comparing the mean number of pages per chapter. On average, each chapter in the MM covers no more than two pages. In the CM, the mean numbers of pages per chapter is seven.

More generally, the MM is less daunting to the reader than the CM when one considers the following statistics: The MM has 3974 words, spread over 10 chapters. The MM is 28 pages long. In contrast, the CM has 11,717 words, spread over six chapters, and is 56 pages long.

Constructivism

It is important to note that many of the minimalist design principles can also be found in a recent approach to instruction called constructivism. Like minimalism, constructivism places a high value on experience-based learning in

context-rich environments. Being problem-oriented, constructivistic environments stimulate students to apply their knowledge and skills almost immediately. There is no advance explanation of the major ideas involved, students have to discover these themselves. Clearly, this capitalises on the learner's prior knowledge and his or her ability to make inferences. Moreover, constructivism emphasises the need to stimulate or support students to follow their own plans and goals (see Duffy and Jonassen, 1991, 1992).

The two approaches thus share important, fundamental principles of instruction. This suggests that minimalism is not just a useful idea for instructing first-time users of (relatively) simple programs. Indeed, as Carroll's (1993) recent work on minimalist instruction for object-oriented programming shows, these principles may also serve instructors for other audiences and in broader contexts with good results.

Experimental investigation of minimalism

This study addresses two important methodological criticisms that have been levelled against the claims of Carroll *et al.* (1987). First, the two manuals were constructed simultaneously as a response to the critique that one of the two manuals in the study might serve as a bridge for getting an improved version of the other one (see Nickerson, 1991, p.25). Second, to enhance the ecological validity of their study, Carroll *et al.* (1987) allowed subjects to consult their manual during the test. Experimentally, this is a problematic procedure because one can ascribe results to a learning effect of the manual, or to its referential qualities. While it seems important to allow manual usage during the test, it is probably wise to study how such consultations affect the results.

This study therefore examines the same hypotheses as Carroll *et al.* (1987), but does so under more rigorous methodological control. That is, subjects working with the MM were expected to complete training faster, to complete more test items successfully, and to commit fewer errors. We also checked whether manual consultations affected these outcomes. In addition, it was predicted that the MM would reduce the need for consultation of the manual after practice.

Method

Subjects

Twenty-five adult volunteers (8 men and 17 women) participated in the experiment. Subjects were randomly assigned to one of two conditions (i.e. types of manual). There were 13 subjects in the MM group and 12 in the CM (control) group. The subjects' mean age was 37, with a range of 17 to 63. As with age, the educational background of the subjects varied considerably (e.g. three subjects had not completed secondary school while two subjects had university degrees). Most subjects had less than ten hours of computer experience, and only two had ever seen a word-processor. Preliminary checks revealed that both groups were equivalent with regard to age, sex, intelligence (as measured on a standardized vocabulary test), educational background, and previous experience with computers.

Procedure

Experimental sessions lasted one working day, with a maximum of 8.5 hours (breaks included). There was one half-hour for initial data gathering, up to six hours (maximally) for practice and one hour to complete the performance test. Only three subjects (1 working with the MM, and 2 working with the control manual) were still processing the last chapter of the manual when practice time was over.

The experiment was introduced as a study 'to get to know how people learn how to work with a word-processor' and subjects were instructed to do as they might do at home. They were instructed to seek help from the experimenter only when they had been stuck for some time and could not solve their problem alone. In addition, the subjects were asked to think aloud during practice.

After practice, the subjects were given the performance test (which they were not told of in advance). They were instructed to try to solve the problems on their own as much as possible, but they were also told that they could consult their manual if they wanted to.

Experimental materials

Manuals. The MM and CM were developed from scratch at the same time. For the construction of the MM the original manual of Carroll *et al.* (1987) served as a reference. For the control manual this reference was provided by a best-selling and prize-winning tutorial. Both manuals were constructed for first-time users learning how to operate a word processor (i.e. WordPerfect 5.1). For obvious reasons, the two manuals were given the same general layout and typography.

Both manuals were submitted to checks by experts. In addition, iterative tests were conducted to improve their qualities. That is, concept versions of the MM and the CM were submitted to audits by three experts (i.e. a linguist, an instructional specialist, and a software expert). In addition, pilot tests were run with subjects from the target audience. Revisions in both manuals were made on the basis of these findings.

Performance test. The performance test was given after practice and was timed. The subjects had to perform several tasks of increasing difficulty. A new task was presented after the subject had completed, or decided to terminate, the previous one.

The test consisted of three types of tasks: managerial, simple word-processing, and complex word-processing tasks. For each file the subjects had to perform two or three basic *managerial* tasks such as saving the file, clearing the screen and retrieving the document. The subjects also had to execute a number of basic tasks in each file. Sixteen of these *simple word-processing* tasks had been treated in the manual. Typical examples included activities such as typing, deleting, copying and moving text, and simple tasks for restyling words and space between lines. Five tasks, all presented in the last two files, measured transfer. These items dealt with *complex word-processing* tasks that subjects had not practised and for which they could not find any reference in the manual (e.g. inserting a hard page-break, or altering a footnote). Since almost none of

the subjects successfully completed the transfer items, the outcomes for these items are not reported.

Coding and scoring of the dependent variables

The *training time* was registered by the observer. It began as soon as subjects started to study their manual and ended when subjects indicated that they were ready.

The number and mean of (*successfully*) *completed tasks* were scored to safeguard against subjects who spent an inordinate amount of time on, mostly, their last, incomplete task.

The following working definition was used to distinguish errors. A subject is committing an error when:

- (1) a value or number is filled in for a wrong system cue (e.g. choosing 'Yes' for page number, when the task requires choosing the right number for the margins);
- (2) a wrong choice is made after a good system cue (e.g. selecting 'No' for saving a file when it has to be 'Yes');
- (3) a wrong menu-choice leads to a non-menu bar screen (e.g. selecting 'Line Spacing' to try to set a word in italics).

The *mean number of errors* was registered.

In addition to recording the *mean number of recoveries*, the *type of recovery* was also noted. Recoveries were coded as system-prompted or self-initiated. System-prompted recoveries are corrections stimulated by cues from the program. For example, the program detects mistakes made while saving a file, prompting the users' corrective actions (e.g. 'Press: Enter' appears on the status-bar). In contrast, self-initiated recoveries are corrections that are the result of the user's own initiative to recover from a mistake. The *time* it took subjects from making a mistake until correction or task progression was registered as error-time.

The *use of the manual* during the test was registered. In addition, the study also examined whether these consultations affected the outcomes on the performance tasks.

Data analyses

Most of the data was analysed by means of (M)ANOVAs, with type of manual as the independent variable. In addition, regression analyses with condition, age, educational background, intelligence and sex as the predictors were performed to see how much of the variance could be accounted for by manual type. The adjusted R^2 statistic, which corrects for sample size and number of variables, is reported. Chi-square tests were used for the qualitative data.

Results

Training Time. MM subjects needed substantially less time to complete their training. MM subjects took 227 minutes on average, whereas CM subjects needed 261 minutes. This difference was *not* significant ($F(1,23)=1.86$,

Table 2. Mean scores of subjects on tasks completed and tasks completed successfully by manual type and task type

	Minimal manual	Control manual
Completed tasks		
Managerial	11.5 (3.9)	6.3 (2.5)
Simple WP	13.4 (2.6)	10.7 (3.5)
Successfully completed tasks		
Managerial	11.3 (3.7)	6.3 (2.5)
Simple WP	10.9 (3.1)	8.1 (3.6)

Note: Standard deviations are given in brackets

The statistics for completed tasks are: MANOVA: $F(2,22)=7.44, p < 0.01$; ANOVA-managerial: $F(1,23)=14.75, p < 0.001$; ANOVA-simple WP: $F(1,23)=4.87, p < 0.05$. The statistics for successfully completed tasks are: MANOVA: $F(2,22)=7.23, p < 0.01$; ANOVA-managerial: $F(1,23)=14.96, p < 0.001$; ANOVA-simple WP: $F(1,23)=4.45, p < 0.05$.

$p > 0.10$). There were large standard deviations; in both conditions some subjects needed twice as much time as others.

(Successfully) Completed Tasks. Subjects working with the MM completed significantly more tasks than subjects who practised with the CM. As can be seen from Table 2 this difference is present for managerial tasks as well as for simple word-processing tasks. Manual type accounted for a significant portion of the variance (27%).

This pattern is maintained when looking at the number of successfully completed tasks (Table 2). MM subjects scored significantly higher for the two tasks and manual type also again accounted for a significant portion of the variance (32%). It is important to note that the subjects in both conditions were highly successful on the performance test, completing successfully about 100% of the managerial tasks and 80% of the simple word-processing tasks that they attempted.

Error Rate, Time and Recovery. There was a marginal multivariate main effect of manual type on error rate ($F(2,22)=3.16, p=0.06$). MM subjects made fewer errors on managerial and simple word-processing (WP) tasks, but this difference approached significance only for the simple WP tasks (see Table 3). Manual type accounted for 9% of the variance.

Manual type also affected error time. Subjects who had practised with the MM spent considerably less time on the detection and correction of errors, especially for simple WP tasks (see Table 4). On average, MM subjects spent 14% of their time for these tasks on error recognition and recovery, whereas CM subjects needed about 35%. Manual type accounted for 26% of the variance.

There was a significant difference in the way in which the subjects tried to correct their mistakes. As Table 5 shows, MM subjects initiated their own error recovery attempts more often than CM subjects who depended more upon

Table 3. Mean number of errors per task as a function of manual type and task type

	Minimal manual	Control manual
Managerial tasks	0.41 (0.49)	0.96 (1.07)
Simple WP tasks	0.57 (0.37)	0.84 (0.36)

Note: Standard deviations are given in brackets.

The statistics are: MANOVA: $F(2,22)=3.16, p=0.06$; ANOVA-managerial: $F(1,23)=2.86, n.s.$; ANOVA-simple WP: $F(1,23)=3.48, p=0.08$.

Table 4. Mean time on errors as a function of manual type and task type

	Minimal manual	Control manual
Managerial tasks	103 (24)	203 (85)
Simple WP tasks	302 (84)	835 (116)

Note: Times are given in seconds, standard deviations are given in brackets.

The statistics are: MANOVA: $F(2,22)=5.26, p=0.05$; ANOVA-managerial: $F(1,23)<1, n.s.$; ANOVA-simple WP: $F(1,23)=10.23, p<0.01$.

Table 5. Mean percentage of self-initiated error recovery strategy of subjects as a function of manual type and task type

	Minimal manual	Control manual
Managerial tasks	10	56
Simple WP tasks	14	45

Note 1: Scores for system-prompted recovery strategy are simply: 100 minus the cell scores (e.g. 90% of the MM subjects used a system-prompted recovery strategy for managerial tasks).

Note 2: Statistics are: Managerial: $\chi^2(1,26)=3.78, p=0.051$; Simple WP: $\chi^2(1,73)=6.24, p<0.05$.

system cues for error detection and recovery. In both cases the relationship between manual type and recovery strategy was fairly strong (Cramer's ϕ' was respectively 0.46 and 0.32 for managerial tasks and simple WP tasks).

Nevertheless, there was *no* effect of manual type on the mean number of successful recoveries. MM subjects as well as CM subjects corrected about 95% of their errors on managerial tasks, and about 50% of their mistakes on simple WP tasks.

Manual Consultations. For both types of tasks MM subjects consulted their manual significantly less often than CM subjects (see Table 6). In both cases the relationship between manual type and recovery strategy was fair (Cramer's ϕ' was respectively 0.27 and 0.23 for managerial tasks and simple WP tasks).

In addition, MM subjects did *not* perform better than CM subjects when consulting their manual (for managerial tasks: $\chi^2(1) < 1, p > 0.10$; for simple WP tasks: $\chi^2(1)=2.69, p > 0.10$). This rules out the possibility that the MM subjects

Table 6. Mean percentage of consultations of manual by manual type and task type

	Minimal manual	Control manual
Managerial tasks	1 (1)	13 (10)
Simple WP tasks	15 (26)	27 (34)

Note: The number of observations is given in brackets.

The statistics are: Managerial: $\chi^2(1)=14.30, p<0.001$; Simple WP: $\chi^2(1)=14.22, p<0.001$.

successfully completed more tasks because of their consultations during the test.

Discussion

In general, the results from this experiment support Carroll *et al.*'s (1987) finding that a minimalist manual significantly and strongly improves the subjects' scores on performance tests. Compared with a control (conventional) manual, the MM helped subjects complete tasks faster and spend less time on errors and error recovery. In addition, the outcomes of recovery strategies and consultations during the test indicate that MM subjects become more independent from their manual.

With its procedure-oriented approach the MM runs a risk of not developing the user's model of the underlying principles of the program enough (cf. Way, 1991). Important signs of a deeper understanding of the program come from the data on error-recovery success and success on transfer items. On both these accounts the MM is not worse, but is also not better than the control manual. The lack of an error-recovery effect which was also absent in the study of Carroll *et al.* (1987) is noteworthy in view of the amount of error information included in the MM.

In view of this and other findings (see Black *et al.*, 1987; Frese *et al.*, 1988; Gong and Elkerton, 1990; Lazonder and Van der Meij, *in press*; Olfman and Bostrom, 1988; Ramsay and Oatley, 1990; Vanderlinden *et al.*, 1988; Wendel and Frese, 1987), it is probably correct to conclude that a minimalist approach to first-time users' documentation is preferable to a more conventional approach. Carroll and his colleagues (1984; 1987; 1990a; 1990b) seem to have laid a good basis for the development of better user documentation. With their radical departure from the then existing conventional manuals they have triggered interesting dialogues and significantly advanced our understanding of certain design principles. Future research should aim to deepen and broaden this understanding.

Acknowledgement

We wish to thank Judith Ramsey and Keith Oatley for constructive criticism of the manuscript.

References

- Allwood, C.M. (1984) 'Error detection processes in statistical problem solving' *Cog. Sci.* 8, 413-437
- Arnold, W.A. (1988) 'Learning modules in minimalist documents' *Proc. 35th Int. Tech. Commun. Conf. (ITCC)* Society for Technical Communication, WE 16-19
- Black, J.B., Carroll, J.M. and McGuigan, S.M. (1987) 'What kind of minimal instruction manual is the most effective?' in Bullinger, H.J., Shackel, B. and Kornwachs, K. (eds) *Proc. 2nd IFIP Conf. Human-Computer Interaction* Elsevier, 159-162
- Carroll, J.M. (1984) 'Minimalist training' *Datamation* 30, 125-136
- Carroll, J.M. (1990a) 'An overview of minimalist instruction' *Proc. 23rd Annual Hawaii Int. Conf. Systems Sciences (IEEE'90)* IEEE Computer Society Press
- Carroll, J.M. (1990b) *The Nürnberg Funnel: Designing Minimalist Instruction For Practical Computer Skill* MIT Press
- Carroll, J.M., Smith-Kerker, P.L., Ford, J.R. and Mazur-Rimetz, S.A. (1987) 'The minimal manual' *Human-Computer Interaction* 3, 123-153
- Carroll, J.M. (1993) 'Techniques for minimalist documentation and interface design' *Proc. Conf. Quality of Technical Documentation* (Enschede, Netherlands)
- Duffy, T.M. and Jonassen, D.H. (1991) 'Constructivism: new implications for instructional technology?' *Educ. Technol.* 31, 7-12
- Duffy, T.M. and Jonassen, D.H. (1992) *Constructivism and the technology of instruction: A conversation* Lawrence Erlbaum Associates
- Frese, M., Albrecht, K., Altmann, A., Lang, J., Von Papstein, P., Peyerl, R., Prümper, J., Schulte-Göcking, H., Wankmüller, I. and Wendel, R. (1988) 'The effect of an active development of the mental model in the training process: experimental results in a word processing system' *Behav. Inf. Technol.* 7, 295-304
- Gong, R. and Elkerton, J. (1990) 'Designing minimal documentation using a GOMS model: a usability evaluation of an engineering approach' in Carrasco Chew, J. and Whiteside, J. (eds) *Proc. CHI'90 Conf.* ACM Press, 99-106
- Hallgren, C (1992) 'The Nürnberg Funnel: a minimal collection' *J. Comput. Documentation* 16, 11-17
- Horn, R.E. (1992) 'Commentary on the Nürnberg Funnel' *J. Comput. Documentation* 16, 3-11
- Lazonder, A.W. (in press) 'Minimalist documentation and the effective control of errors' in Jansen, C., van der Poort, P. Steehouder, M.F. and Verheijen, R. (eds) *Quality Of Technical Documentation* Rodopi
- Lazonder, A.W. and Van der Meij, H. (1992) 'Towards an operational definition of the minimal manual' *Technical Report No. IST-MEMO 92.02* University of Twente, Enschede, the Netherlands
- Lazonder, A.W. and Van der Meij, H. (in press) 'The minimal manual: is less really more?' *Int. J. Man Mach. Studies*
- Nickerson, R.S. (1991) 'A minimalist approach to the "paradox of sense making"' *Educ. Researcher* 20,9, 24-26
- Olfman, L. and Bostrom, R.P. (1988) 'The influence of training on use of end-user software' *Proc. Conf. Office Inf. Systems* ACM Press, 110-118
- Ramsay, J.E. and Oatley, K. (1991) 'Designing minimalist tutorials from scratch' *Instructional Science* 21, 85-97
- Tripp, S.D. (1990) 'Book review of: *The Nürnberg Funnel*' *Educ. Technol. Res. and Dev.* 38, 3, 87-90

- Vanderlinden, G., Cocklin, T.G. and McKita, M.** (1988) 'Testing and developing minimalist tutorials: a case study' *Proc. 35th Int. Technical Communications Conf. (ITCC)* Society for Technical Communication, 196-199
- Van der Meij, H.** (1992) 'A critical assessment of the minimalist approach to documentation' *SIGDOC'92 Conf. Proc.* ACM Press, 7-17
- Way, E.C.** (1991) *Knowledge representation and metaphor* Kluwer
- Wendel, R. and Frese, M.** (1987) 'Developing exploratory strategies in training: the general approach and a specific example for manual use' in **Bullinger, H.J., Shackel, B. and Kornwachs, K (eds)** *Proc. 2nd IFIP Conf. Human-Computer Interaction* Elsevier, 943-948