The Effects of Screen Captures in Manuals: A Textual and Two Visual Manuals Compared

Abstract—This study examined the use of screen captures in manuals. Three designs of manuals were compared, one textual and two visual manuals. The two visual manuals differed in the type of screen capture that was used. One had screen captures that showed only the relevant part of the screen, whereas the other consisted of captures of the full screen. All manuals contained exactly the same textual information.

We examined the time used on carrying out procedures (manual used as a job aid) and the results on retention tests (manual used for learning). We expected to find a trade-off between gain in time and learning effects. That is, we expected that higher scores on the retention tests involved an increase in time used and, vice versa, that gains in time would lead to lower retention test scores. We also explored the influence of manual design on user motivation.

For job-aid purposes, there were no differences between manuals. For learning, the full-screen captures manual and the textual manual were significantly better than the partial-screen captures manual. There was no proof for the expected trade-off. More learning was not caused by an increase in time used. We found no effects on user motivation.

This study does not yield convincing evidence to support the presence of screen captures in manuals. However, if one wants to include screen captures, this study gives clarity for the type of screen capture to choose. The use of full-screen captures is preferable to partial ones. Finally, we conclude that documentation designed to expedite the execution of tasks does not necessarily hamper the learning that may result.

Index Terms—Documentation, motivation, screen captures, usability, visualizations.

Nowadays, the use of visuals in user manuals for the computer industry seems to be a must. Designers devote much time and energy to creating attractive manuals. Often this is done by including various screen captures throughout the manual. These screen captures are presented for more than merely a decorative function. They can show, for example, a required start-screen or the correct result of an action. Designers face important questions such as when (for which type of information) and
which screen captures (full or partial) to use in their documentation.

Handbooks on technical documentation reveal very little about the use of screen captures. Price and Korman [1] treat the topic in one paragraph, stating that screen captures should be used for two purposes: 1) to show the results of action steps taken and 2) to show the object to act upon in the next action step. The only design guideline they offer is to use callouts to draw the users’ attention to key parts of a window. Similarly, in Dynamics in Document Design, Schriver [2] does not discuss the role and design of using screen captures in technical documentation. She just gives several general guidelines on combining the use of words and graphics in document design.

The most extensive discussion on screen captures comes from Horton [3]. Among other things, Horton questions whether screen captures always have a purpose that justifies their cost. Horton also mentions that screen captures offer visual relief on pages full of text and states that “when used appropriately and placed wisely, they make procedures easier to learn and quicker to follow” (p. 148). What actually is “appropriate” and “wise” is described in three guidelines (p. 148):

1) In tutorials, screen captures should be offered to let the user imagine how to use the system.
2) Screen captures should be used to let the user verify the display, especially when the target group is the novice computer user.
3) If only part of the screen is important, only that part should be shown. The pages “should not be cluttered with what the users already know.”

In short, research and advice about the use of screen captures in technical documentation are limited. The questions when and which screen captures to use in manuals are, for the bigger part, unanswered by the literature.

The “when” question concerns the types of information whose presentation can be supported by the use of screen captures. A common and valuable classification into types is the distinction between conceptual and procedural information. Conceptual information offers explanations and supports goal setting. Procedural information supports direct or indirect user actions and can be divided into action information, error information, and coordinative information [4]. Screen captures can be used to support the presentation of all these information types, for example, by showing a target screen (goal setting), the outcome of an action step (action information), or a specific button (coordinative information).

The question “where” to use screen captures is about the appropriate place of a screen capture on a manual page. Screen captures can be placed on the left of the text, on the right, or in the flow of the text.

Asking “which” screen captures to use often boils down to asking whether to use full- or partial-screen captures. Should the designer present everything that is shown on the screen, or is a display of only the relevant part better? The main difference between full- and partial-screen captures concerns the use of context. Full-screen captures show the complete interface. Partial captures show little (e.g., the active window) or no context (e.g., a single button).

So far, we have only talked about design issues of screen captures in manuals. However, our primary drive to focus on screen captures is that we think they can improve documentation. They can, for example, support locating a specific menu or object and make checking the correctness of a screen easier. Using screen captures for such specific goals will facilitate a bridge between what is written in the manual and what is seen on the computer screen. Documentation can be used in mainly two ways: for learning how to work with a program and for carrying out tasks. Improving documentation therefore means two things: to speed up task execution and to improve learning.

Whether the presence of screen captures speeds up task execution is a question that has been studied by Van der Meij [4]. In a study comparing a visual and a textual manual, he found a significant positive effect of screen captures on task execution time. He offers three explanations for this. One, the connection between what is written and what is shown on the screen is now presented in a single source: the manual. Users may thus have fewer difficulties in handling the two separate sources. Two, there is no need for the user to translate the text into an image because the way it should look is already printed in the manual. Three, fewer switches between manual and screen are needed. Because of the screen captures, the manual becomes more self-contained. In general, these arguments all share the core idea that screen captures reduce coordination problems.

Van der Meij [4] also mentions some drawbacks of screen captures. One of these is user passivity. The presence of the screen captures may discourage users to study the interface and reduce the need for users to search and examine the screen very closely. Another drawback is that the redundancy of both screen captures and text may be disadvantageous because the user has to process the same information twice, which imposes an undesirable heavy cognitive load.

This raises the intriguing question of the existence of a trade-off. Is what is gained from using screen captures for speeding up task
execution also a loss for learning? When screen captures reduce cognitive effort and speed up task execution, they may simultaneously fail to maximally activate the user in using and exploring the interface, and thereby fail to support learning. In other words, users benefit from the manual as a job aid but suffer a loss for learning due to decreased cognitive effort. To give a specific example, when a screen capture in a manual is used to support locating a button on the interface, the user will be quicker in finding that button than without that screen capture. In the meantime, there is no need to search the interface for the relevant button. Consequently, the user will gain less knowledge of the interface as a whole.

Thus, it seems fair to predict that screen captures in manuals cannot serve both goals: to speed up task execution and simultaneously, to improve learning. For this study, a main question is whether this prediction holds. We predict that faster training leads to lower learning. In addition, we look at a design issue of screen captures. More specifically, we examine the role of full-screen captures versus partial-screen captures.

Three manuals (tutorials) were compared: a textual manual (Text), a manual supporting procedural information with partial-screen captures (V-Part), and a manual that supported procedural information with full-screen captures (V-Full). The textual manual was designed according to minimalist principles and heuristics [5] and formed the basis for the two visual manuals. Partial-screen captures were added to the action steps in the V-Part manual, whose design was inspired by the Visual Learning Guide manuals by Gardner and Beatty [7]. A full-screen capture showed the complete interface. Example pages of the three manuals can be found in Appendix A (Text), B (V-Part), and C (V-Full).

The main goal of the study was to find out if these manuals have a different effect on speed of task execution and learning, and whether faster training leads to lower learning.

We expected that the manual with full-screen captures would lead to the quickest task execution. Because of the lack of visual support in the textual manual, we expected this one to be the slowest. For learning, we expected the opposite. As users of the textual manual were expected to devote the most effort on getting to know the system, the largest effect on learning was expected there. For users of the manual with full-screen captures the need to actively examine the system was expected to be the lowest. Consequently, learning effects were expected to be worst for that manual. We expected the manual with partial-screen captures to take the middle position for both speed on task execution as well as learning effects.

We examined two levels of learning: learning to perform the same tasks as trained with the manual (trained tasks) and tasks that were different than trained with the manual (untrained or transfer tasks). For example, a manual can contain information on how to make a bulleted list: the trained task. Matching untrained tasks can be making a bulleted list in multiple levels or making a numbered list.

To examine the effects of manual type on the job-aiding purpose of the manual, we measured training time. The total training time consisted of time that users needed to read explanations, carry out procedures, and explore the program. To find out whether the visual manuals gave visual relief and were viewed as more attractive, user motivation was measured.

**Method**

**Subjects** Seventy-three Dutch students from the Faculty of Educational Science and Technology participated in the experiment. The mean age of the experimental group was 21.2 years (SD = 2.4 years). The subjects were classified as intermediate or experienced computer users on the basis of their score on the Computer Self-Efficacy Scale questionnaire. It was expected that subjects with less computer experience would benefit more from screen captures than would experienced users. Subjects were randomly assigned to one of the three experimental conditions: Text, V-Part, or V-Full. Table I shows how the subjects were distributed. Classification of subjects into levels of computer experience served two additional purposes. One, it made clear that the subjects’ level of experience was average or above average: they were certainly not beginners. Two, it made it possible to check that subjects were indeed randomly distributed over the three conditions.

**Materials**

**Computers:** The sessions were held in a computer room with 20 IBM compatible Pentium Pro 166 computers with 32 MB of RAM. During the experiment, all subject actions with the computer program were logged automatically.

**SimQuest and Motion Application:** Subjects learned to use the SimQuest authoring tool version 1.1 [8]. SimQuest uses an object-oriented approach, which means that a collection of ready-made elements can be used to create an application or program. With SimQuest, the teacher or designer creates a learning environment
that offers a set of simulations, assignments, and explanations that enable learners to explore a specific domain. A main component in the subject’s education (educational science and technology) is learning how to systematically design instruction using various media. As SimQuest is a state of the art tool for designing multimedia instruction, it was expected that the subjects would be very interested in learning to use it.

The SimQuest application used to exemplify the creation of a simulation environment in the manuals dealt with the physics domain of motion. The application lets the students explore the relationship between initial velocity, velocity at a certain point and time, and acceleration. Subjects are shown various simulations with moving motorcycles, trains, cars, scooters, and others. Assignments make it possible for the student to check the correctness of any discovered relationship. Explanations such as videos and textual information introduce and discuss the variables used in the simulations and assignments. Main tasks trained in the manual concerned modifying and creating simulations, assignments, and explanations.

Manuals: The manuals were written in English, and all contained exactly the same text. In order to avoid differences in reading, we attempted to keep the layout of the three manuals as similar as possible. Even so, the presence of screen captures led to manuals of different sizes. The text manual consisted of 32 pages. The V-Part had 54 pages containing a total of 231 partial-screen captures, and the V-Full had 58 pages, containing a total of 87 full-screen captures.

Each chapter in the manual consisted of two sections (see Fig. 1): a guided section with a brief task description and detailed action steps to accomplish the task, and an exploratory section, which offered

---

**Table I**

**Distribution of Subjects Per Condition**

<table>
<thead>
<tr>
<th>Manual</th>
<th>Intermediate (m/f)</th>
<th>Experienced (m/f)</th>
<th>Row total (m/f)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>10 (0/10)</td>
<td>15 (3/12)</td>
<td>25 (3/22)</td>
</tr>
<tr>
<td>V-Part</td>
<td>10 (1/9)</td>
<td>15 (5/10)</td>
<td>25 (6/19)</td>
</tr>
<tr>
<td>V-Full</td>
<td>9 (1/8)</td>
<td>14 (6/8)</td>
<td>23 (7/16)</td>
</tr>
<tr>
<td>Column total</td>
<td>29 (2/27)</td>
<td>44 (14/30)</td>
<td>73 (16/57)</td>
</tr>
</tbody>
</table>

m = male, f = female

---

Fig. 1. Example of a guided section (‘Adding interface elements’) and an exploratory section (‘Try it yourself’) in the manual.
one or more exercises comparable to the task practiced in the guided section. In line with the minimalist approach [5], these exploratory sections are an important feature in the manual.

**Questionnaires and Tests:** The subjects received a questionnaire with general questions about gender, age, and previous experience with authoring tools. Nineteen participants (7 males and 12 females) stated that they had used an authoring tool at least once. In addition, there were 20 questions to classify the subjects as intermediate or experienced computer users. For this purpose, the Computer Self-Efficacy Scale [9] questionnaire was translated into Dutch. This questionnaire used a five-point agree–disagree scale.

An electronic questionnaire, based on Keller’s ARCS theory [10], asked the subjects about their motivational state. The four motivational elements from the ARCS theory (Attention, Relevance, Confidence, and Satisfaction) were captured in four SimQuest-specific questions, which were shown (every 15 min) in an automatically appearing window (see Fig. 2). Subjects were asked to answer the questions by moving the sliders, which always displayed the middle, neutral, position when presented.

Two tests were used to determine learning effects: an immediate test and a delayed test. The items in the tests had two levels of difficulty:

- Items that measured trained tasks (exercises that were the same as practiced with the manual); and
- Items that measured untrained, also known as transfer tasks (new tasks that were different from practiced tasks).

Table II shows the number of test items in the immediate and delayed test.

**Procedure** The experiment consisted of three sessions: practice, an immediate test, and a delayed test. Before the practice session, the subjects answered the questionnaire on gender, age, previous

---

**Fig. 2.** Pop-up motivation questionnaire.

**TABLE II**
**NUMBER OF ITEMS IN IMMEDIATE AND DELAYED TEST**

<table>
<thead>
<tr>
<th></th>
<th>Immediate test</th>
<th>Delayed test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trained</td>
<td>Untrained</td>
</tr>
<tr>
<td>Number of test items</td>
<td>18</td>
<td>23</td>
</tr>
</tbody>
</table>
experience with authoring tools, and computer experience.

The practice session lasted 4 hours maximally. It was held from 9:00 A.M. to 1:00 P.M., with two coffee breaks of 15 min. At the start of this session, the subjects were told that their task was to learn how to work with SimQuest. They were told to work on their own, using only the manual for support. During practice, every 15 min a pop-up screen appeared asking the subjects the four questions about their current motivational state. The subjects could stop practicing when they felt they were able to comfortably use SimQuest.

The immediate test session took place the same day, starting at 2:00 P.M., and lasted a maximum of 2 hours. The subjects were asked to try to do their best without the use of a manual. They were also told that some things in the test would be rather different from what they had practiced that morning. They were further told that this session would end at 4:00 P.M., but that they could leave when they were finished.

The delayed test session took place one week after the first test session. The subjects could work a maximum of 2 hours on this test. As with the immediate test, they were not allowed to use their manual.

Coding and Scoring

Computer Experience, Gender, and Previous Use of Authoring Tools: The questionnaire on computer experience used a five-point disagree-agree scale. Subjects with a mean score lower than 3 were classified as intermediate users, subjects with a score of 3 or higher as experienced users (see Table I). Female subjects were scored as 1, and male subjects as 2. Subjects who stated that they had used an authoring tool at least once before were scored as 1 and subjects that never used an authoring tool before as 0. Computer experience, gender, and previous use of authoring tools were all variables at a nominal level.

Time: During practice, all subjects' actions were logged. These logs allowed us to determine training time for guided and exploratory sections. Time used for coffee breaks was subtracted.

Time used on the guided parts showed a direct effect of manual type on task execution. It showed how long subjects took to complete the reading of the short explanations and to carry out the action steps. Time used on exploratory parts showed the time users spent in exploration. Both in guided and exploratory sections, subjects had to save their work as a last action. Saving was therefore taken as the transition to the next section.

A MANOVA showed no significant relations between time and computer experience, time and gender, or time and previous use of authoring tools. Therefore, there was no need to correct for these three variables when examining differences on time.

Motivation: The data of the motivation pop-up questionnaire consisted of a maximum of 12 repeated measures. The first measurement was removed because it was used for practice. After the ninth measurement, the number of subjects that answered the questionnaire dropped below the pre-set criteria of 85% (it was 84%). Therefore, only measurements 2–9 were used in the analysis. Examination of the instruments' reliability showed that the questionnaire was highly reliable (see Table III).

Tests were performed to determine if the results on the four measures could be combined into one value indicating the subjects' motivational state. Table IV shows the correlations between the four items. Pearson correlations indicate that the four indeed share a fundamental basis. Therefore, the scores for the four measures were combined into a composite score for motivation.

A MANOVA showed no significant relations between motivation and gender, and motivation and previous use of authoring tools. A significant relation was found between motivation and computer experience ($F(1, 72) = 4.60, p < 0.05$). Computer experience will therefore be treated as a covariate when testing for

<table>
<thead>
<tr>
<th>TABLE III</th>
<th>RELIABILITY OF MOTIVATION MEASURES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cronbach's Alpha</td>
<td></td>
</tr>
<tr>
<td>Attention</td>
<td>0.95</td>
</tr>
<tr>
<td>Relevance</td>
<td>0.96</td>
</tr>
<tr>
<td>Confidence</td>
<td>0.94</td>
</tr>
<tr>
<td>Satisfaction</td>
<td>0.95</td>
</tr>
<tr>
<td>Motivation (combination of the four factors)</td>
<td>0.96</td>
</tr>
</tbody>
</table>
differences on motivation between manuals.

Learning Effects: For each trained or untrained item, a subject could receive a score of 1 if the item was performed correctly or a score of 0 if executed incorrectly.

A MANOVA showed no significant relations between learning effects and computer experience, learning effects and gender, or learning effects and previous use of authoring tools. Therefore, there was no need to correct for these three variables when examining differences on learning.

Results

Time: Table V shows the means and standard deviations of the time users spent in dealing with the guided and exploratory sections of the manuals.

No statistically significant differences for practice time between manuals were found on guided sections of the manuals. One explanation is that all texts provide sufficient coordinative information. The action steps clearly explain what to do and where to act. The screen captures may therefore have been redundant, offering no vital or new information. Inaccurate or unclear screen captures may even lead to confusion, and consequently to delay. Another reason might be the transparency of the interface. The interface may have been so easy to use, that (extra) coordinative information was not necessary at all. Yet a third explanation may lie in the specific content of the guided sections. These sections contain procedural (doing) as well as conceptual (reading) information, and the recorded time reflects the processing of both information types. Clearly, this somehow moderates any time gain of screen captures because they mainly support the handling of procedures. A better view of the effects of screen captures on time requires a filtering out of all reading time.

The three conditions differed considerably in the time subjects spent on exploratory sections. Subjects with the text manual spent almost twice as much time exploring the program as did users of V-Part manual. This difference was statistically significant $(F(2, 72) = 3.37, p < 0.05)$; with a Tukey HSD-test at 0.05). Users of the text manual thus appeared more willing to devote time on trying things themselves than users of the V-Part manual. This may signal a difference in motivation, although this could not be proven statistically (see next section). The statistically significant difference on time used on exploratory sections continues to exist when time on exploratory sections is taken as a proportion of the total training time $(F(2, 72) = 3.39, p < 0.05)$; with a Tukey HSD-test at 0.05). The mean proportion of time used on exploratory sections of the total training time varied between 0 and 24%.

Motivation: An ANCOVA with computer experience as covariate showed no significant effect of manuals on motivation $(F(2, 72) = 0.78)$, Experienced users were more motivated. Regression analysis showed that 7% of the variance on motivation could be explained by computer experience $(F(1, 71) = 5.29, p < 0.05)$.

Despite the fact that there were no statistically significant main effects of manual type on motivation, the results consistently favor the V-Full manual (see Table VI). These results give an indication of the possible visual relief that this type of manual is supposed

---

**TABLE IV**

<table>
<thead>
<tr>
<th></th>
<th>Attention</th>
<th>Relevance</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relevance</td>
<td>0.41*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confidence</td>
<td>0.58*</td>
<td>0.73*</td>
<td></td>
</tr>
<tr>
<td>Satisfaction</td>
<td>0.47*</td>
<td>0.55*</td>
<td>0.74*</td>
</tr>
</tbody>
</table>

* $p < 0.001$

---

**TABLE V**

<table>
<thead>
<tr>
<th></th>
<th>V-Part</th>
<th>Text</th>
<th>V-Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guided sections</td>
<td>7583 (1071)</td>
<td>7483 (1243)</td>
<td>7427 (1184)</td>
</tr>
<tr>
<td>Exploratory sections</td>
<td>596 (597)</td>
<td>1133 (834)*</td>
<td>936 (767)</td>
</tr>
<tr>
<td>Total</td>
<td>8179 (1104)</td>
<td>8616 (1169)</td>
<td>8362 (1223)</td>
</tr>
</tbody>
</table>

* $p < 0.05$ compared to V-Part
to offer when compared to the text manual.

Examinations of the results in the course of time showed results that pointed in the same direction, favoring the V-Full manual. Fig. 3 shows the flow of measurements on motivation. It can be seen that the V-Full manual is the best motivator on all factors, all the time, but not significantly so. A repeated measures test found no proof in favor of one of the three manuals \( F(2,60) = 0.596 \).

Surprisingly, there are no clear differences between the V-Part and text manuals. In other words, there seems to be no extra benefit in offering partial-screen captures in comparison to plain text. Indeed, there may be an opposite effect. As motivation slightly drops over time using the V-Part manual, it may well be that partial-screen captures tend to de-motivate.

**Learning Effects:** All subjects, regardless of the manual with which they had practiced, performed quite well on the items that measured trained tasks. On the immediate test as well as the delayed test, more than 87% of the tasks were performed correctly (see Table VII). This ceiling effect is troublesome because it strongly limits the chances of finding any significant differences on trained tasks.

The untrained tasks were performed somewhat less well (see Table VIII). Both the V-Full and the text manual outperformed the V-Part manual on the delayed test \( F(2,71) = 5.55, p < 0.01 \), with a Tukey HSD-test at 0.05. The difference between V-Full and V-Part suggests that the V-Full users have gained a better understanding of the program. Explaining the difference between the text and V-Part manuals is more difficult. One account may be that the text manual forces

---

**TABLE VI**

Means (Standard Deviations) of Motivational Factors (Scale 0–100, Default Score Was 50)

<table>
<thead>
<tr>
<th>Order of conditions</th>
<th>V-Part</th>
<th>Text</th>
<th>V-Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attention</td>
<td>64.21 (25.36)</td>
<td>67.59 (20.44)</td>
<td>70.65 (18.84)</td>
</tr>
<tr>
<td>Relevance</td>
<td>64.31 (22.49)</td>
<td>66.19 (13.57)</td>
<td>69.38 (13.84)</td>
</tr>
<tr>
<td>Confidence</td>
<td>67.98 (11.58)</td>
<td>68.91 (17.17)</td>
<td>72.23 (10.98)</td>
</tr>
<tr>
<td>Satisfaction</td>
<td>62.69 (9.95)</td>
<td>63.02 (19.92)</td>
<td>66.86 (11.99)</td>
</tr>
<tr>
<td>Motivation</td>
<td>65.11 (17.05)</td>
<td>66.16 (11.21)</td>
<td>69.79 (11.94)</td>
</tr>
</tbody>
</table>

Fig. 3. Development of motivation in the course of time where high scores indicate high motivation. (The neutral [default] score was set at 50.)
users to more actively explore the program. The results on time, where significantly more time was spent on exploratory parts by text users than V-Part users, supports this explanation. Apart from devoting more time, it could also be that the partial-screen captures interfere with understanding the program. On the one hand, the information given by the partial-screen captures may have been too limited to support users to learn to understand the program. On the other hand, the partial-screen captures may have confused users who actively constructed their own understanding of the program and therefore disturbed that construction process.

*Trade-Off Between Time and Learning Effects:* Examinations of Pearson correlations between training time and learning effects revealed an intriguing pattern (see Table IX).

The correlations for total training time show that there is a negative relationship (immediate test) or no relationship (delayed test) with the scores on the retention tests. This means that shorter training time leads to higher test scores respectively, that there is no relationship between training time and test scores. Correlations on guided sections are all negative, except for the V-Full manual on the delayed test, where the correlation is nil. From theory, it was expected that a gain in time would work against learning. Therefore, we expected the correlations to be positive. Instead, the results show that shorter training time leads to more learning, and longer training time leads to less learning. These findings clearly contradict a trade-off between training time and learning.

It is interesting to see that there were no main differences between manuals in this respect. Manuals like these, designed—among others—to shorten training time do not obstruct learning. On the contrary, there is some indication that users benefit.

This finding made us reconsider the need to take training time into consideration as a correcting factor when considering effects.
of manuals on learning. An ANCOVA with total training time as covariate still showed a significant effect of conditions for untrained test items on the delayed test ($F(2, 71) = 5.23, p < 0.01$). In other words, time did not interact with the main effect found for learning.

**Conclusion**

The experiment does not make it perfectly clear whether screen captures are a necessary feature for the improvement of documentation. Looking at the results, there is proof that a design in which partial-screen captures are coupled to action steps is not a good solution. On several measures, the subjects who had worked with the V-Part manual performed worse than the other subjects.

When the V-Full and text manuals are compared, there is no proof that one leads to more learning than the other. Also, in time used on guided and exploratory parts, no differences were found between the V-Full and text manuals. The use of full-screen captures suggests a motivating influence. The experiment has not proven this assertion, however.

Another important finding of this study is that the use of screen captures does not lead to a trade-off between gain of time and benefits for learning. The results show that better performances on the tests cannot be asserted to an increase in training time. Therefore, it can be concluded that documentation designed to expedite the execution of tasks does not necessarily hamper the learning that may result.

One might conclude that devoting much time and resources on presenting screen captures in manuals is not worth the effort. A closer look at the experiment caution against such a conclusion. There are several arguments to show that it may be too early to tell.

An important premise for this experiment was that we wanted to have a situation that was close to reality. It can be argued that learning how to use a computer program with a manual as the only source of information, and for 3 hours in a row, is a not realistic situation. Learning a computer program at home or at work may go quite differently. It may take four half-hour sessions over a period of two weeks instead of one long session. Using a visual manual instead of a textual one in this case, where you have to restart several times, may then have its benefits.

A final consideration is that the subjects who participated in this experiment may not represent computer users in general. As students and our faculty must and do use computers quite a lot, their computer experience (and level of formal educational training) is probably higher than that of average computer users. For real novices, differences between using textual and visual manuals may again be stronger.

Looking at the types of manual used in the experiment, a few remarks can be made. It was surprising to see that there were no differences in time on task, especially because Van der Meij [4] found quite strong effects. This may very well be explained by the manuals used in both experiments. Van der Meij used manuals that were meant for job-aiding purposes only. That documentation consisted almost completely of procedural information whereas the tutorial in this experiment was a balanced combination of conceptual and procedural information. As the focus of a tutorial is primarily on learning, or better, in getting to understand the program by doing, reading, and exploring, less gain in time can be expected.

A second remark pertains to the difficulty of the tests in combination with the quality of the manual. The results for learning show that subjects were very capable of performing the tasks on which they had trained and even on those they did not, both in the immediate and the delayed test. These results indicate that the tests may have been too easy or that the manual did its job well in teaching the subjects how to use the program. Too well, perhaps? If this is indeed the case, benefits from screen captures can only be small.

Also, the two designs of the visual manuals were in a way unsophisticated. In each manual, only one type of screen capture design was used, full or partial. These screen captures should support various user activities. For example, a screen capture may focus the user’s attention at the start of a procedure, may help the user in identifying and locating screen objects during task execution, and may ease verifying a screen state at the end of a procedure. Functions such as these may require different screen capture designs. A visual manual in which screen captures are presented in a way that their roles and designs are optimally attuned to one another may function much better than the visual (or textual) manuals tested in this study. Van der Meij and Gellevij [11] have proposed a framework for research in this fashion. That framework, created after we completed this study, distinguishes four roles and four design dimensions for screen captures in manuals. With this framework, we think it is possible to fine-tune roles and designs for screen captures in a more sophisticated way.

Manuals quite often not only serve the purpose of instructing and supporting users. Visually attractive manuals can very well be part of the selling strategy for software packages which could be a legitimate reason to choose a visual manual. If the marketing department makes that decision, they should, as this experiment suggests, opt for presenting full-rather than partial-screen captures.
APPENDIX A:
TEXT EXAMPLE PAGE

The SERVIVE project  SIMQUEST V-September

Adding explanations to the application
Before you can start editing your explanation, you must:
• drop it from the library and drop it into your application
• give it a meaningful name.

Dragging and dropping an explanation
You already decided which explanation element you are going to use, you can take it from the library and put it into your application.

1 In the Library window, select Video
2 Drag Video from the Library window and drop it into the tabsheet level 1 of the Application window

Naming an explanation
To keep it clear what your elements contain, you can give the explanation a meaning full name. The video fragment this explanation is going to contain is about a motorbike.

1 In level 1 of the Application window, select Video
2 Click your right mouse button and choose Rename
3 Type Motorbike, and click OK

Editing an explanation
To be able to create or edit your explanation you should:
• open the explanation editor,
• specify the content, and
• specify the learner description.

Opening the explanation editor
First you have to open the explanation editor.

1 In level 1 of the Application window, select Motorbike
2 Click your right mouse button and choose Edit

Check if the editor appears on the screen.

Specifying the content
You use the Specification tabsheet to specify which video file must be presented.

1 Select the Specification tabsheet
2 Click Select video
3 Select the folder motion.res
4 Select the subfolder video
5 Choose Optr-rem.avi and click OK
6 Click Apply

Specifying the learner description
You can offer your student a short description of the explanation’s content that they can read before they decide to start the explanation. This description is called the learner description. The learner description will be shown to
**APPENDIX B: V-PART EXAMPLE PAGE**

**The SERVIVE project**

Adding explanations to the application

Before you can start editing your explanation, you must:
- drop it from the library and drop it into your application
- give it a meaningful name.

Dragging and dropping an explanation

You already decided which explanation element you are going to use, you can take it from the library and put it into your application.

1. In the *Library window*, select **Video**

2. Drag **Video** from the *Library window* and drop it into the tabsheet *level 1* of the *Application window*

Naming an explanation

To keep it clear what your elements contain, you can give the explanation a meaning full name. The video fragment this explanation is going to contain is about a motorbike.

1. In *level 1* of the *Application window*, select **Video**

2. Click your right mouse button and choose **Rename**

3. Type: **Motorbike**, and click **OK**

---

*Modifying and Creating explanations*
APPENDIX C: V-FULL EXAMPLE PAGE

Adding explanations to the application

Before you can start editing your explanation, you must:
- drop it from the library and drop it into your application
- give it a meaningful name.

Dragging and dropping an explanation

You already decided which explanation element you are going to use, you can take it from the library and put it into your application.

1. In the Library window, select Video.
2. Drag Video from the Library window and drop it into the tabsheet level 1 of the Application window.

Naming an explanation

To keep it clear what your elements contain, you can give the explanation a meaning full name. The video fragment this explanation is going to contain is about a motorbike.

1. In level 1 of the Application window, select Video.
2. Click your right mouse button and choose Rename.
3. Type: Motorbike, and click OK.

Modifying and Creating explanations
ACKNOWLEDGMENT

We would like to thank D. Farkas for his valuable help on the study and the review of this paper.

REFERENCES


Mark Gellevij works at the Department of Instructional Technology of Twente University, Enschede, The Netherlands. After graduating from this department, he worked as a technical writer in trade and industry. He is now working on his doctoral thesis about designing user support for complex software environments. His paper “Screen Captures in Software Documentation,” coauthored with Hans van der Meij, has recently been selected as the winner of an award of Distinguished Technical Communication in the Society for Technical Communication’s outstanding article award competition for 1998.

Hans van der Meij works at the Department of Instructional Technology of Twente University, Enschede, The Netherlands. His main areas of interest are information seeking behavior and technical documentation. In the latter area, he has published 16 chapters and 33 articles. One of his articles, coauthored by J. M. Carroll, received an Award for “Distinguished Article of 1995” from the Society for Technical Communication. Hans also received the “Best Transactions Paper Award for 1997” from the IEEE Professional Communication Society. His studies in technical documentation combine theory, practice, and empirical research. He currently conducts inventories on the roles and designs of screen captures, procedures, warnings, and indexes.
**Ton de Jong** is Professor of Instructional Technology at the University of Twente, Enschede, The Netherlands. He received the Ph.D. degree from the Eindhoven University of Technology, Eindhoven, The Netherlands, on the topic “Problem solving and knowledge representation in physics for novice students.” He specializes in problem solving and discovery learning with computer simulations.

**Jules M. Pieters** is Professor of Instructional Technology (appointed 1991), Faculty of Educational Science and Technology at the University of Twente, Enschede, The Netherlands. After graduating in Experimental Psychology at the University of Nijmegen, he wrote his doctoral thesis about the role of psychomotor control and language in spatial localization (1980). He joined the faculty in Twente in 1980. Currently, he is Dean of the Faculty of Educational Science and Technology. Pieters is codirector of the research program in Cognitive Tools and Instructional Design. He is involved in research projects on knowledge acquisition and transfer and on designing learning environments.