

Visuals in Instruction

Mark Gellevij

2002

Ph.D. thesis
University of Twente



Twente University Press

Also available in print:

www.tup.utwente.nl/catalogue/book/index.jsp?isbn=9036517451

Visuals in Instruction

ico



Twente University **Press**

Publisher: Twente University Press, P.O. Box 217, 7500 AE Enschede, the Netherlands,
www.tup.utwente.nl

Cover design: Jo Molenaar, [deel 4] ontwerpers, Enschede
Print: Océ Facility Services, Enschede

© M.R.M. Gellevij, Enschede, 2002

No part of this work may be reproduced by print, photocopy or any other means
without the permission in writing from the publisher.

ISBN 9036517451

VISUALS IN INSTRUCTION

FUNCTIONS OF SCREEN CAPTURES IN SOFTWARE MANUALS

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. F.A. van Vught,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op donderdag 6 juni 2002 te 16.45 uur

door

Mark Remko Michiel Gellevij
geboren op 20 juli 1972
te Hengelo

Dit proefschrift is goedgekeurd door de promotoren

prof.dr. A.J.M. de Jong

prof.dr. J.M. Pieters

en de assistent-promotor

dr. H. van der Meij

Acknowledgments

About a week after the interview, I received a phone call from Ton de Jong to inform me that if I was interested, the Ph.D. position was mine. I was somewhat taken aback, as in my view, the conversation illustrated that my competencies to fulfill this job were as yet limited. I would like to thank Hans, Ton, and Jules for offering me the opportunity to write this thesis. Their support has been continuous and strong, each in his own characteristic way. And although working closely together with three such intelligent and devoted people has its occasional difficulties, it is something I certainly treasured.

They warned me during the interview, that working on a thesis can be a lonely business. I have a lot of people to thank for preventing this from happening. Fellow Ph.D. students and colleagues, our secretaries, members of the Serve project, workers at Tolab, and students that participated in the studies all made the work definitely enjoyable. Moreover, their help was crucial for the realization of this thesis. I am most grateful for all their effort and support.

I would like to thank my parents as well as Manon's for convincing me that taking the Ph.D. position would be the right thing to do. I would like to express my gratitude to family and friends for showing their interest in my work over the years and, more importantly, for providing me with an environment in which I was able to feel secure and happy. Thanks also go to Jan for putting things in perspective by frequently asking when my 'proefwerk' would be ready, and when we would finally have my 'spreekbeurt'.

Finally, I thank Manon who claims that she has not contributed to this thesis in any way. It is beyond doubt that she is completely wrong.

Mark Gellevij
April 2002

Table of Contents

1 INTRODUCTION	1
Rationale and development of scientific knowledge	1
Development of scientific knowledge	2
Explore	3
Theorize	3
Validate	4
Instructional design	4
The use of software manuals	5
The potential of visuals in instruction	6
Text-picture processing	10
Organization according to the rationale	12
Overview of the book	13
References	14
2 EXPLORATION	19
Introduction	20
Method	23
Subjects	23
Materials	23
Computers	23
SimQuest and Motion application	23
Manuals	24
Questionnaires and tests	24
Procedure	26
Coding and scoring	26
Computer experience, gender, and previous use of authoring tools	26
Time	27
Motivation	27
Coding and scoring - Learning effects	28
Results	28
Time	28
Motivation	29
Learning effects	31
Trade-off between time and learning effects	32
Conclusion	33

Acknowledgement	35
References	35

3 FOUR FUNCTIONS OF SCREEN CAPTURES IN SOFTWARE MANUALS **37**

Introduction	38
Switching attention	40
Developing a mental model of the program	42
Verifying screen states	46
Identifying and locating window elements and objects	49
Conclusion	52
References	53
Figure references	55

4 VALIDATION: SWITCHING ATTENTION **57**

Introduction	58
Method	61
Participants	61
Materials	61
Manuals	61
Questionnaires and tests	62
Procedure	63
Coding and scoring	63
Number of participants in the analyses	63
Switching attention	63
Learning effects	64
Results	64
Check for randomization	64
Training time and Learning effects	64
Switching attention	65
Discussion	66
Acknowledgement	68
References	68

**5 VALIDATION: DEVELOPING A MENTAL MODEL;
IDENTIFYING AND LOCATING WINDOW ELEMENTS;
VERIFYING SCREEN STATES** **69**

Introduction	71
Processing text and pictures: Dual Coding	71
Processing text and pictures: Cognitive Load	72
Processing text and pictures: Cognitive Load or Dual Coding?	73
Instructional Functions of Screen Captures	74
Method	76
Participants	76
Materials	76
Computers	76
SimQuest and Motion application	76
Manuals	78
Questionnaires and tests	82
Procedure	84
Coding and scoring	84
Number of participants in the analyses	84
Computer experience, age, gender, topic of study, and grade level	85
Cognitive load	85
Training time on manual chapters	85
Verification	85
Learning effects	86
Results	86
Check for randomization	86
Cognitive load	87
Training time	87
Verification	89
Learning effects	90
Discussion	90
References	94

6 VALIDATION: VERIFICATION OF SCREEN STATES **97**

Introduction	98
Method	100
Participants	100
Materials	100
Computers	100
SimQuest	101

Manuals	101
Questionnaires and tests	103
Procedure	104
Coding and scoring	104
Number of participants in the analyses	104
Gender, age, and computer experience	105
Time	105
Progress checks and error recovery	105
Cognitive load	105
Opinions	106
Results	106
Check for randomization	106
Time and problem solving	106
Cognitive load	108
Opinions	110
Discussion	110
References	111
7 CONCLUSIONS AND DISCUSSION	113
Instructional Design: Functions of screen capture functions	113
Applying functions and a genre approach	114
Reasons for the distinction of screen capture functions	115
Relations between screen capture functions	116
Text-Picture Processing: Dual Coding and Cognitive Load	117
Suggestions for future studies	119
Methodological issues	119
Research topics	120
References	121
NEDERLANDSE SAMENVATTING (Dutch summary)	123
APPENDICES	
A: Text Example Page for Chapter 2	
B: Visual-Part Example Page for Chapter 2	
C: Visual-Full Example Page for Chapter 2	
D: Design dimensions	



Introduction

This book is about the use of visuals to support learners in their process of learning. The type of *instruction* that is used is a paper manual meant to learn how to use a (complex) software program. The specific type of *visuals* to support the learning process are pictures of (parts of) the computer screen. These visuals will further be called screen captures. The goal of this book is to provide insight into when and how screen captures in software manuals influence the learner (e.g., motivation and cognitive load), learner behavior (e.g., speed of task performance and interaction between the manual and the computer screen), and outcomes of learning (e.g., learning effects).

The papers presented in chapters 2 through 6 hold studies that are taken from published and submitted articles. This introductory chapter serves as an organizer for these studies. Organization is offered in two ways. First, a description of the rationale for studying visuals and the development of scientific knowledge are discussed and connected to the chapters of this book. Second, an overview is given by presenting short summaries of the succeeding chapters.

Rationale and development of scientific knowledge

The rationale for studying screen captures in software manuals is derived from two perspectives. The first perspective is that of instructional design, which aims at understanding instruction and providing guidelines for the design of instructional materials. The goal of instruction is supporting people to acquire knowledge and skills. This is done by assisting learners with registering information, processing information in short-term memory, encoding information for storage in long-term memory, and retrieval of information from long term memory (Bruning, Shaw, & Ronning, 1995). Understanding these information processing procedures, specifically for the processing of text-picture combinations, is the second perspective of studying screen captures in software manuals.

This section discusses the instructional design and text-picture processing perspective of the rationale. It starts with a short outline of how scientific knowledge

can be acquired as this represents the overall structure of the book. The section ends with situating the various studies within the two perspectives.

Development of scientific knowledge

According to Aristotle, science is making common-sense knowledge explicit in a two-phase process. It starts with observations from practice and the articulation of explaining principles for these observations. This phase is called induction. Then, from the explaining principles, hypotheses are derived meant to test if the principles indeed explain the original observation. This second phase is called deduction (De Vries, 1984).

Warries and Pieters (1992) apply this to research on instructional systems. In the inductive phase, learners in an instructional system are observed. Based on the registered observations, assumptions are formulated about the working of the instructional system. From these assumptions, a tentative theory is derived about supposed patterns and laws in the instruction. In the deductive phase, hypotheses are deduced to test whether the theory indeed represents reality. That is, whether applying the patterns and laws in the instructional design make the instruction work.

Although the above process consists of two phases (induction and deduction), it actually contains three steps: (1) doing initial research, (2) formulating a theory, and (3) doing research to test the theory. The research carried out in the first and third step differ fundamentally. The initial research is of an explorative nature. It is carried out based on curiosity about a phenomenon in relation with certain variables (Van der Ploeg, 1994). Research in the third step is meant to test or validate a theory. The development of scientific knowledge can thus be described in three parts: exploring, theorizing, and validating. Figure 1-1 shows these parts.

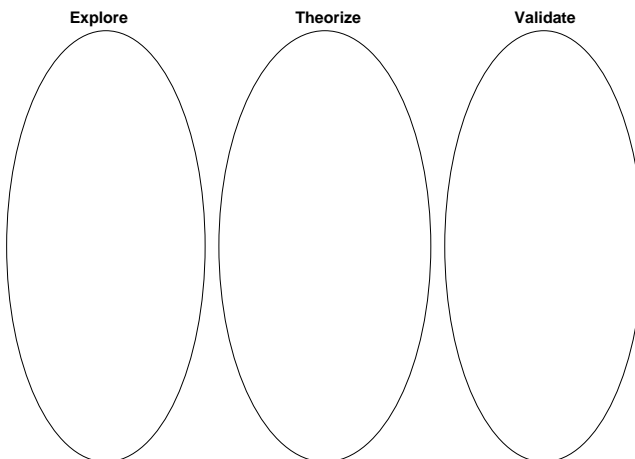


Figure 1-1. Three phases in development of scientific knowledge

Explore

The development of scientific knowledge starts with curiosity. The urgency to figure out how things work is the basis for the active development of all types of knowledge, not only scientific knowledge. The method in which knowledge is acquired gives scientific knowledge its scientific hallmark. Knowledge is called scientific when it is obtained in a systematic, precise, accurate, and objective manner (Garmonsway, 1970). The first step in developing scientific knowledge is thus to translate initial curiosity into measurable matters and using a controlled and provable method to satisfy this curiosity.

In case of this book, curiosity about the use of screen captures in software manuals comes from the observation that (1) most contemporary manuals include screen captures, and (2) people are quite willing to buy 'visual' manuals. Apparently, designers of software manuals assume that using screen captures improves their instruction, and consumers find these books appealing. In the exploration phase, the use and effects of two commercially successful visual manuals and a textual one are compared to find out if visual manuals are indeed an improvement for learning and motivation.

Theorize

Based on findings in the explorative phase, a first 'controlled' impression of the topics under study is obtained. The effects and relations of the variables that are examined can be described and these findings can subsequently be combined into an understanding of the subject matter. Although the method used is as objective as possible, there will often be a need to make an interpretation of the experimental results. Understanding of the subject matter is therefore a combination of objective findings and reasoning.

Based on the explorative study comparing the three manuals, two important interpretations were made that initiated the composition of theory. First, it looked as if screen captures were put in the manual regardless of the learner's task. There seemed to be no connection between the instruction and the place and form of screen captures. This conclusion led to the necessity of theory in which screen captures are consciously used to support the learner while carrying out the various tasks that comprise the process of learning to use a computer program, and where the design of the screen capture is maximally tuned to (one of four) instructional (supportive) functions. These functions were derived from an analysis of the user's tasks while working with the software manual and computer simultaneously. Based on this analysis, screen captures are believed to support the switching of attention between the paper manual and the computer screen, development of a mental model of the software program, verification of screen states, and identification and localization of window elements and objects. (A detailed description of the task-analysis and the four functions is given in Chapter 3.)

Second, it became clear that it would be unwise to consider only the functions of screen captures without taking into account the processing of information. Testing a variety of screen capture designs while ignoring the cognition of the user would lead to multi-interpretable or even incomprehensible findings. Theories on Dual Coding

and Cognitive Load were used to gain insight in the way learners deal with multimodal instruction, in this case instruction holding text-picture combinations.

Validate

After formulating a theory it is necessary to test whether the assumptions and predictions of this theory are correct. It needs to be examined whether the theory indeed represents reality.

Experimental studies in the validation phase are identical to studies in the exploration phase with respect to their requirements such as the need to follow a systematic approach, being precise, accurate, and objective. A difference between studies in both phases is that in the validation phase there will be (stronger) expectations for certain results. After all, the theory provides predictions for certain behavior, relations, and effects.

In this book, the theoretical notions were validated by comparing visual manuals with textual ones. The screen captures in the visual variants were deliberately applied to support one or more of the four identified instructional functions. The designs of these screen captures were matched to their function. Cognitive load was measured to examine which of two theories, Dual Coding theory or Cognitive Load theory, would best explain the process of text-picture processing. The expectations were that using screen captures as instructional support would lead to improvements in training time and learning while cognitive load would *not* negatively influence the learning process.

The above three phases in which scientific knowledge is acquired serve as the general approach of this book. Within this approach, two main perspectives, that of instructional design and of text-picture processing, determined the content of the studies.

Instructional design

Research in instructional design examines the structure and effects of instruction. Such research leads to descriptive theories about the working of instructional systems. Instructional design also contributes to the design of new instruction and improvements in existing instruction. Instructional design theory intends to prescribe optimal methods of instruction that offers explicit guidance to help people learn and develop, and to bring about desired changes in knowledge and skills (Reigeluth, 1983, 1999).

The common definitions for instruction are rather broad. Instruction is described as: “intentional facilitation of learning toward identified learning goals” (Smith & Ragan, 1999, p. 2), or “realization, by means of a method, that a learner, within a system, under certain conditions, reaches a beforehand defined learning goal” (Warries & Pieters, 1992, p. 18). Consequently, the form of instruction can be very diverse, ranging from traditional classroom settings in which a teacher addresses a group of students, to individualized interactive multimedia packages. In this book, instruction takes the form of a paper manual. The topic of instruction in these manuals is learning to use a (complex) software program. The intended contribution

of this book to the field of instructional design is the development of empirically validated design rules and guidelines for the use of visuals, in this case screen captures, in software manuals.

The use of software manuals

There seems hardly any necessity to state that computers are used by a majority of people. That the use of computers is well integrated into our day-to-day lives, for both professional as well as for 'fun' purposes, goes almost without saying. Despite this integration, working with a computer or computer based application at first almost never goes impeccably. Although computer programmers and interface designers strive after user interfaces that are as intuitively as possible, some sort of instruction nearly always appears to be required to learn how to work with it. Producers of computer hardware, software, and computer applications therefore most often offer some sort of user support to their customers. The usual types of support for these products are manuals and online help.

Recent studies disrupt the myth that manuals are not used. Contrary to public belief, the use and popularity of manuals seems to surpass online help. Jansen and Balijon (2001) asked 201 Dutch respondents whether they read the manual after buying appliances like a VCR or mobile phone. Only 8% never read the manual. Of the 92% that read the manual, 38.8% stated that they always, 32.3% often, and 20.9% sometimes read the manual. As a second question they asked the respondents how they read the manual. Twenty percent answered that they read it cover-to-cover, 54% read it in scanning mode, 23% in case they got stuck, and 3% answered to never read it¹. Shriver (1997) found similar results for American participants. She asked (also 201) respondents: 'Generally speaking, how do you read instruction manuals?' (p. 213). Only 4% never read the manual. Fifteen, 46%, and 35% read their manuals cover-tot-cover, in scanning mode, or as a reference respectively. Vromen and Overduin (2000) found similar results asking users of Davilex-software about the user support offered. Of the 224 respondents, 95.5% stated that they used the manual. Another question in their study was what they would do in case of encountering a problem. Over half of the respondents would use a manual as the first form of support to solve their problem, despite the availability of online help. DeTienne and Smart (1995) interviewed 400 people about their use of manual and online help. The manual was used by 95% of the respondents, online help by 60%. Of these, 65% used the manual once every two weeks or more, compared to 46% using online help in that frequency.

From these findings it can be concluded that manuals are used as the main form of support when learning how to work with a computer. Considering the widespread and extensive use of computers and computer appliances, this makes examining ways to improve instruction by software manuals worthwhile.

¹ It is remarkable that the percentage of respondents that states to never read the manual is not about the same when asked these two questions. Both findings do however consistently deny the assertion that hardly anyone reads a manual.

The potential of visuals in instruction

Instructional material in which the use of various media is integrated improves learning more than instruction with only one medium (Mayer, 1999; Mayer & Sims, 1994; Mayer, Moreno, Boir & Vagge, 1999; Mayer & Gallini, 1990; Robinson, Robinson, & Katayama, 1999; Kulhavy, Stock, Peterson, Pridemore, & Klein, 1992). Especially Mayer and his co-authors have repeatedly shown that users benefit from a multimodal approach, whose most common form is that of a mixture of words and pictures. Using software manuals in which screen captures are combined with text are therefore expected to be an improvement over purely textual manuals. Before the start of the research reported in this book, there were three studies in which this specific type of instruction (paper software manuals), using these specific type of pictures (screen captures) are examined.

Nowaczyk and James (1993) compared three manuals: a textual one, a manual including button and icon information, and a manual with button and icon information and full screen captures. They examined time needed to accomplish a task, retention of knowledge of the screens, and appreciation of helpfulness of the actions in the manual. The task covered by the manual was transferring a file from an IBM computer to a Macintosh. The authors found that using the manual with button and icon information lead to the fastest task performance, whereas subjects using the textual manual performed slightly worse. The manual with buttons, icons and screen captures lead to worst performance and lowest appreciation of helpfulness. There were no differences on the retention tests. Users stated that they found it important to receive support in locating the relevant part of the screen to act on. This seems to contradict to their low appreciation of the manual with the full screen captures as this full screen manual offered the most support on locating.

Sweller and Chandler (1994) compared a text plus screen capture manual *without* using a computer with a textual manual *with* the use of a computer. The task was learning how to use a spreadsheet. Training time was shorter and learning was better for the text plus screen capture condition. Although the findings of this study are in favor of the use of screen captures in manuals, such a conclusion should be treated with caution. In the visual condition, participants could *not* interact with the computer program. The question remains if such interaction would have deteriorated or further improved learning and speed of task execution.

Finally, Van der Meij (1996) compared a textual manual with a full screen captures manual. He examined time and accuracy, which was measured as number and type of errors made. The task was working with a database on coatings with a non-graphical (line based) computer program. A large time gain was found for the visual manual. No differences in accuracy were found.

The above studies were meant to examine the influence of screen captures on variables as training time and learning. Table 1-1 summarizes the findings of the three studies.

Table 1-1. Comparing results of studies on the use of screen captures in manuals

Authors	Manuals	Dependent Variables	Significant results
Nowaczyk and James (1993)	Text Icons and buttons Full screens with icons and buttons	Time Learning Appreciation	Full screens slowest Full screens worst appreciated
Sweller and Chandler (1994)	Text Visual	Time Learning	Screens quickest Screens best for learning
Van der Meij (1996)	Text Full screens	Time Accuracy	Full screens quickest

It is hard to obtain unequivocal guidelines for the design and use of screen captures in manuals from this summary. Reasons for these contradictory findings may not only be caused by the presentation and form of screen captures (icon, object, partial, or full screen captures). When comparing the above studies, other noteworthy differences can be observed: the type of manual (job-aid vs. tutorial), the magnitude of the task (short and simple tasks vs. long and complex tasks), the type of computer program (Graphical User Interface (GUI) vs. line based), the experience of the user group (novice vs. experienced users), and the use of the actual computer during training. These differences may also very well account for the variations of findings so far.

Apart from these differences, all studies have one important characteristic in common. Screen captures used in the instructions of the reported studies all use a particular screen capture design (icon, object, partial, or full screen captures) consistently throughout the manual without much variation. The screen captures thereby do not take into account an instructional function, or support that is needed at a particular moment in the execution of the users' task. Peeck (1993) provides an overview of potential effects of picture processing in instructional texts from the viewpoint of such instructional functions or, as he calls them, instructional interventions. This overview consists of an hierarchy of instructional interventions coupled to possible effects of processing pictures and the risk of pictures being ignored (see Table 1-2).

A screen capture is a specific type of picture used in an instructional intervention. Categorization of the function of such a screen capture in Peeck's hierarchy leads to an indication for the probability that a screen capture will indeed be used and processed. Table 1-3 illustrates this by coupling instructional interventions to the four screen capture functions. From this categorization, it appears that the expected effects for the processing of screen captures in software documentation are likely to differ depending on their specific instructional function. In turn, it is likely to expect that a picture that has a high chance of being processed also has a high influence on the process and outcomes of learning.

Table 1-2. Instructional interventions, their possible effects on picture processing, and risk of ignoring the picture (Adapted from Peeck, 1993)

Instructional intervention	Possible effect on picture processing	Risk of user ignoring intervention
Ask or tell user to pay attention to illustration	Small	Large
Tell user what to observe an illustration <i>in general</i>	Rather small	Rather large
Tell user what to observe in a <i>particular</i> picture	Moderate	Moderate
Tell student to do something with illustration (compare, trace, locate, label, complete) <i>without</i> controllable product	Rather large	Rather small
Tell student to do something with illustration (compare, trace, locate, label, complete) <i>with</i> controllable product	Large	Small

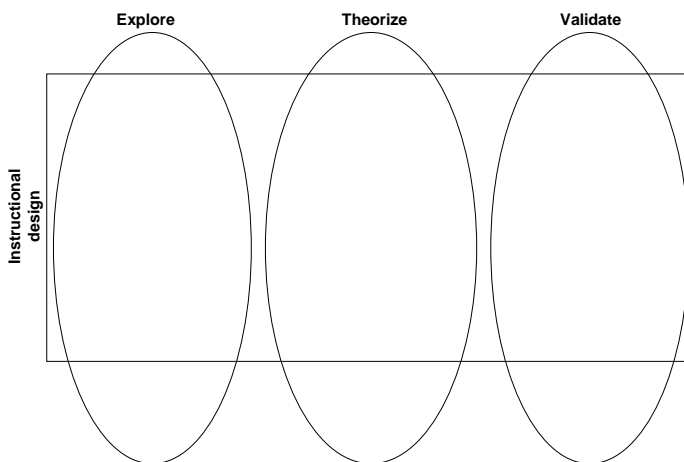
Screen captures used to support *switching attention* are meant to prompt the user to look up from the manual to the screen. For *developing a mental model*, screen captures are meant to impart the look-and-feel of the computer program. Both functions are intended to support the user by *particularly* making clear what to look for. Following Peeck’s view, effects of processing such visually driven instructional interventions are expected to be *moderate*. Screen captures to *identify*, *locate*, and *verify* screen elements, objects, or states are meant to *use* the illustration to label (*identify*), trace and locate (*locate*), and compare (*verify*). While doing this, the actual computer screen is used as the controllable product. Possible effects of processing such interventions are expected to be *large*.

In conclusion, research on the use of text-picture combinations in instruction shows that users should benefit from such an approach when compared to a text-only instruction. Moreover, two of the three available studies on the use of screen captures in software documentation show potential for a visual approach. Finally, Peeck’s view on picture processing suggests that moderate to large effects may be expected from using screen captures for specific instructional functions.

Table 1-3. Instructional interventions coupled to screen capture function and its possible effects on picture processing

Screen capture function	Instructional intervention	Possible effect on picture processing
Switch attention between manual and screen	Tell user what to observe in a <i>particular</i> picture	Moderate
Develop a mental model of the program	Tell student to do something with illustration (compare, trace, locate, label, complete) <i>with</i> controllable product	Large

In this section, it was argued that, from the perspective of instructional design, it is relevant to examine the use of screen captures in software manuals. The main arguments that show such relevance are: (1) the fact that software manuals are the most used form of instruction to learn how to work with a computer program, (2) the opportunity to improve instruction by using visuals, and (3) the limited amount of specific research on this topic. To be able to arrive at a validated theory on functions of screen captures in software manuals, all three steps in scientific knowledge development are needed to go through. Figure 1-2 visualizes the perspective of instructional design in combination with the three phases to develop scientific knowledge.

**Figure 1-2.** The instructional design perspective combined with the three phases of scientific knowledge development

Text-picture processing

Information processing theories describe and hypothesize about processes in the mind when storing and retrieving knowledge and skills. Such processes are described in so-called information processing models of which the common features are joined in the Modal model (Bruning, Shaw, & Ronning, 1995). This model divides human cognition in three memory-systems: sensory memory, short-term memory (also called working memory), and long-term memory. Sensory memory very briefly stores information that comes to us through our senses. That information is next processed in short-term memory, to ultimately be encoded and stored in long-term memory.

The bottleneck in this process seems to be short-term memory because of its limitations. It can hold information for rather short intervals (about a minute), and its storage capacity is limited to 7 plus or minus 2 items or elements. This stands in contrast to long-term memory, which can hold information perhaps for life, and has an enormous capacity (Miller, 1956; Gleitman, 1986).

Two specific theories on information processing focus on short-term memory when text and pictures are to be processed simultaneously. These theories are of particular interest since they seem to give contradictory predictions about the influence of instruction on learning when text and pictures are combined. The theories are: Dual Coding theory and Cognitive Load theory.

According to Dual Coding theory, the basic reason why better learning occurs when words and pictures are combined, lies in how the learner processes information in working memory. Dual Coding, or dual processing, presupposes that when both verbal and visual information is presented, these two types of information are connected in working memory, and this referential connectivity, in turn, contributes to the construction of a strong mental model (Paivio, 1990; Mayer, 1999). Support for these assumptions can be found in the so-called ‘contiguity effect’, which means that text and visuals that are presented in a coordinated way lead to better learning results than text and visuals that are presented separately (see e.g., Mayer & Sims, 1994). In later work Mayer and Moreno (1998) go a step further and propose that working memory consists of two distinct systems: a verbal and a non-verbal system. Verbal information only concerns narrative, and both visual and textual information are seen as belonging to visual information. According to Mayer and Moreno (1998) textual and visual information are initially both represented in the non-verbal system. The textual information is then translated in auditory form and further processed in the verbal system. The research presented in this book started from this assumption that text and pictures are being processed in two distinct but connected working memory systems. In Chapter 7, we will give a further discussion on this.

Cognitive Load theory builds on the idea that there is only one working memory with limited capacity. When people face a task that is already quite difficult, ‘additional’ instructions may therefore be more of a burden than an aid. An example of a ‘burden’ is that learning to use a computer program requires the use of more information sources and the handling of more devices than just the instructional material. Users must not only process the manual, they must also attend to the keyboard, mouse, and computer screen. Cognitive Load theory indicates that this

situation poses two potential risks: redundancy and split attention effects (Chandler & Sweller, 1991; Sweller, 1994; Sweller & Chandler, 1994).

Redundancy occurs when the manual presents screen captures that the user can also see on the computer monitor. The *redundancy hypothesis* predicts that offering the same information twice, as in the case of a depicted computer screen, requires the user to also process this information twice. This double processing takes up unnecessary memory space, space that could have been used for processing other information.

Split attention effects occur when people must attend to different sources of information simultaneously. The *split attention hypothesis* predicts that in such a situation learning is hampered when these sources must be integrated. According to Sweller and Chandler, split attention effects obstruct learning when a text fragment is needed to understand a picture and vice versa because the user then has to process two distinct information sources at the same time.

In conclusion, according to Cognitive Load theory, the use of text-picture combinations in a manual while working with the computer at the same time, is *not* a preferable instructional format. The screen captures are redundant because the same information is already presented on the computer screen. Therefore, they increase the load on working memory, and are thus likely to increase training time and decrease learning. In contrast, Dual Coding theory predicts that instruction with text-picture combinations exceeds text-only information. Thanks to the distinctly different nature of text and pictures and their separate processing of content similar information, the information is better integrated, resulting in a stronger mental model and, more generally, strengthened learning.

The above two theories about the way users process information while working with a combination of textual and visual information are based on an ample research base. To follow the full path of scientific knowledge development seems therefore unnecessary. Theorizing based on exploratory research results is superfluous as the required theories are already available and directly applicable. The exploration phase for the information processing perspective can therefore be omitted. From the perspective of text-picture processing, the start of our examinations is in the theorizing phase with describing the theories and giving hypotheses about the application of these theories when using screen captures in software manuals. Figure 1-3 shows the integration of the perspectives of text-picture processing and instructional design within the three phases of the development of scientific knowledge.

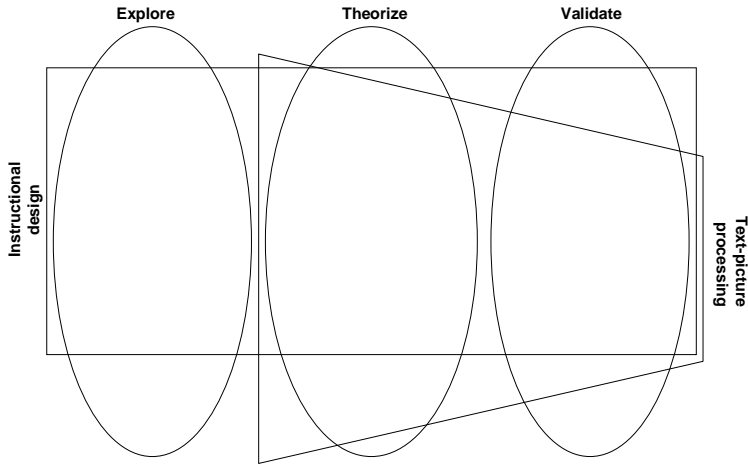


Figure 1-3. The text-picture processing perspective combined with the instructional design perspective and the three phases of scientific knowledge development

Organization according to the rationale

So far, the two perspectives of the rationale for this book (instructional design and information processing) have been explained, as well as the three phases (exploration, theorizing, and validation) in which scientific knowledge can be developed. Figure 1-3 shows the integration of these two perspectives with the three scientific knowledge development phases. This integration serves as an organizer for this book. Figure 1-4 shows how the various chapters fit into this organizer.

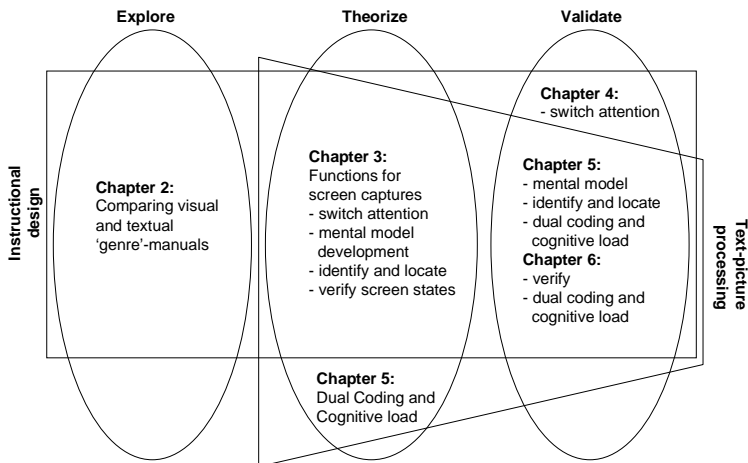


Figure 1-4. Organization of chapters within the rationale

As Figure 1-4 shows, the start in the exploration phase comes from an instructional design perspective. The explorative study is described in Chapter 2.

As a result of this exploration, a theory on the use and design of screen captures in software manuals was constructed. In the theorizing phase, Chapter 3 describes four instructional functions of screen captures to support learning to work with a complex computer program by means of a paper manual. This theory integrates the instructional design and information processing perspective. A description on theories about text-picture processing is presented in Chapter 5.

The Chapters 4, 5, and 6 represent the validation phase. Chapter 4 validates the instructional screen capture function ‘switching attention between manual and screen’. Although this function is a product of both perspectives, the study in this chapter focuses mainly on the instructional design perspective. The remaining three screen capture functions: ‘development of a mental model of the computer program’, ‘identifying and locating interface elements and objects’, and ‘verifying screen states’, are validated in Chapters 5 and 6. The two studies presented in these chapters also explicitly focus on the text-picture processing of instructions. Both instructional design and text-picture processing perspectives are therefore considered to be combined in these chapters.

Until now, this introductory chapter has offered a rationale for the contents given in this book. The organization of, and relations between the chapters are explained based on this rationale. This introductory chapter concludes with summaries of the chapters to give more detailed insight in the goals and setup of the experimental studies executed in the exploration and validation phases, and the theories constructed and presented in the theorizing phase.

Overview of the book

This section gives an overview of the book by presenting short summaries of the successive chapters.

Chapter 2 presents an explorative study in which the use and effects of two visual manuals and a textual (control) manual are compared. The main difference between the two visual manuals is the use of partial screen captures in Stuur’s ‘Windows for children’ manual, and of full screen captures in Gardner and Beatty’s ‘Guided Tour’ manuals. We have called the design of such manuals a genre-based approach. That is, the screen capture design chosen (in this case either partial or full screen captures) is used consistently throughout the manual. Variables that are examined in this study are: training time, learning effects on trained and untrained test items, and motivation. The main conclusion of this study is that designing a visual instruction following a genre-based approach may have its value, yet leaves room for improvement.

Chapter 3 presents four instructionally based functions for the use of screen captures in manuals. The general assumption is that screen captures in a manual can be used to support the user while he or she learns to work with a computer program.

During this process of learning, users face several types of tasks. A task analysis reveals that four types of these task can be made easier by bringing in screen captures. These user tasks are: (1) switching attention between the manual and the computer screen, (2) developing a mental model of the computer program, (3) verifying screen states, and (4) identifying and locating window elements and objects. The important difference between the genre-based approach, used in the visual manuals in the explorative study, and the current instructional approach based on functions of screen captures, is that screen captures are now used in an instructionally driven and goal-based manner.

Chapter 4 reports the results of a study in which the first function for screen captures in software manuals, switching behavior between the manual and the computer screen, is observed. Two variants of a visual manual and one textual manual are compared on the number of times users interact between the manual and the computer screen.

The study in Chapter 5 intends to validate the screen capture functions mental model development, identification and localization, and verification. The use and effects of a visual and textual manual are compared. To test the development of a mental model, learning effects on trained and untrained problem solving tasks are measured. Identification and localization of window elements and objects are tested with items on trained and untrained identification/localization tasks. Verification is tested by measuring time needed to accomplish verification tasks, and by measuring the amount of mistakes that are made while verifying. This study also examines the influence that the use of visuals in instruction has on text-picture processing. It addresses the aforementioned contradictory predictions from Dual Coding theory and Cognitive Load theory.

The study in Chapter 6 addresses some of the methodological difficulties of studying the verification function of screen captures in the previous study. It was carried out to validate the benefits of using screen captures to support the verification of screen states. Again, a visual and textual manual were compared. Users were observed while carrying out a number of short tasks. They received a manual meant to check (verify) why a certain application did not work. Variables in this study were verification time and verification quality (the ability to solve problems). Once more, cognitive load was measured. For this, two different measurements were used: the usual self-scoring and a secondary task-measure on response times.

Chapter 7 reconsiders the theoretical approaches that were presented and used in light of the findings of the studies performed. It ends with offering suggestions for continuing research on functions of screen captures in software manuals and text-picture processing.

References

Bruning, R.H., Shaw, G. J., & Ronning, R. R. (1995). *Cognitive Psychology and Instruction (2nd Edition)*. Englewood Cliffs, New Jersey: Prentice Hall.

- Chandler, P., & Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognition and Instruction*, 8, 293-332.
- DeTienne, K., & Smart, K. L. (1995). The battle of paper documentation versus online documentation. In: *Disappearing borders. Proceedings of Forum95* (November 13-15, 1995). Dortmund: International Council for Technical Communication (INTECOM), 1-3.
- De Vries, G. (1984). *De ontwikkeling van wetenschap [The development of science]*. Groningen: Wolters-Noordhoff.
- Garmonsway, G. N. (1970). *The Penguin All English Dictionary*. Paris: Bordas.
- Gleitman, H. (1986). *Psychology (2nd edition)*. New York: Norton.
- Jansen, C., & Balijon, S. (2001). Hoe worden handleidingen gebruikt en gewaardeerd? [How are manuals used and valued?]. *Tijdschrift voor Taalbeheersing*, 23, 132-144.
- Kulhavy, R. W., Stock, W. A., Peterson, S. E., Pridemore, D. R., & Klein, J. D. (1992). Using maps to retrieve text: a test of conjoint retention. *Contemporary Educational Psychology*, 17, 56-70.
- Mayer, R. E. (1999). Designing instruction for constructivist learning. In: Reigeluth, C.M., *Instructional-Design theories and models Volume II: A new paradigm of instructional theory*. Mahwah, New Jersey: Lawrence Erlbaum Associates.
- Mayer, R. E., & Gallini, J. K. (1990). When is an illustration worth ten thousand words? *Journal of Educational Psychology*, 82, 715-726.
- Mayer, R. E., & Moreno, R. (1998). A split-attention effect in multimedia learning: Evidence for Dual processing systems in working memory. *Journal of Educational Psychology*, 90, 213-330.
- Mayer, R. E., Moreno, R., Boir, M., & Vagge, S. (1999). Maximizing constructivist learning from multimedia communications by minimizing cognitive load. *Journal of Educational Psychology*, 91, 638-643.
- Mayer, R. E., & Sims, V. K. (1994). For whom is a picture worth a thousand words? Extensions of a Dual-Coding theory of multimedia learning. *Journal of Educational Psychology*, 86, 389-401.
- Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63, 81-97.

Nowaczyk, R. H., & James, E. C. (1993). Applying minimal manual principles for documentation of graphical user interfaces. *Journal of Technical Writing and Communication*, 23, 379-388.

Paivio, A. (1990). *Mental Representations: A Dual Coding Approach*. New York: Oxford University Press.

Peeck, J. (1993). Increasing picture effects in learning from illustrated texts. *Learning and Instruction*, 3, 227-238.

Reigeluth, C. M. (Ed.). (1983). *Instructional-design Theories and Models: An Overview of their Current Status*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Reigeluth, C. M. (Ed.). (1999). *Instructional-design Theories and Models (Volume II): A New Paradigm of Instructional Technology*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Robinson, D. H., Robinson, S. L., & Katayama, A. D. (1999). When words are represented in memory like pictures: evidence for spatial encoding of study materials. *Contemporary Educational Psychology*, 24, 38-54.

Schrifer, K. A. (1997). *Dynamics in document design. Creating texts for readers*. New York: Wiley.

Smith, P. L., & Ragan, T. J. (1999). *Instructional Design (2nd edition)*. New York: Wiley.

Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and Instruction*, 4, 295-312.

Sweller, J., & Chandler, P. (1994). Why some material is difficult to learn. *Cognition and Instruction*, 12, 185-233.

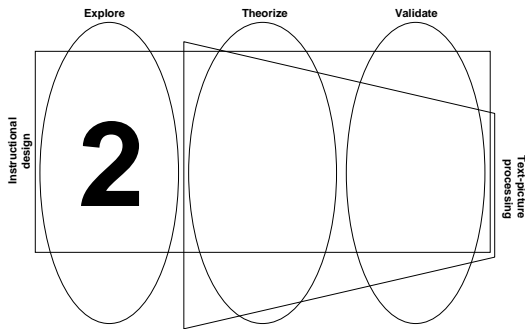
Van der Meij, H. (1996). A closer look at visual manuals. *Journal of Technical Writing and Communication*, 26, 371-383.

Van der Meij, H., & Jenné, M. A. (1999). *Designing and testing visual manuals for the adult learner* (Report for the American Society for Technical Communication (STC)). Enschede, The Netherlands: Twente University, Faculty of Educational Science and Technology.

Van der Ploeg, P. A. (Ed.). (1994). *Geschiedenis en Filosofie van de Sociale Wetenschappen [History and Philosophy of Social Sciences]*. Leiden: Stenfert Kroeze.

Vromen, N., & Overduin, M. (2000), Handleiding: de Titanic onder de gebruikersondersteuning? [Manual: the Titanic of user support?]. *Tekst[blad]*, 6, 42-46.

Warries, E., & Pieters, J. M., (1992). *Inleiding Instructietheorie [Introduction to Instructional Theory]*. Amsterdam/Lisse: Swets & Zeitlinger.



Exploration¹

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.

¹ Gellevij, M., Van der Meij, H., De Jong, T., & Pieters, J. (1999). The effects of screen captures in manuals: Textual and visual manuals compared. *IEEE Transactions on Professional Communication*, 42, 77 - 91.

Introduction

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 (1993) 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 (1997) 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 (1993). 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 is limited. The questions when and which screen captures to use in manuals are, for the bigger part, unanswered by 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 (Van der Meij, 1998). 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 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 that 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 (1998). 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 like 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 (1998) 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 between 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, at the same time 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 mean time, 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 the both goals: to speed up task execution, and to improve learning simultaneously. 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 the minimalist principles and heuristics (Van der Meij & Carroll, 1995) 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 Stuur's visual steps approach (Stuur, 1996). A partial screen capture showed that part (or parts) of the screen the user needed to perform the action step. Examples of partial screen captures are: menus, dialogs, or parts of windows. Full screen captures were added to the action steps in the V-Full manual, whose design was inspired by the 'Visual Learning Guide' manuals by Gardner and Beatty (Gardner & Beatty, 1994). 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 of 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 2-1 shows how the subjects were distributed. Classification of subjects into levels of computer experience served two additional goals. 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.

Table 2-1. Distribution of subjects per condition

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

m = male, f = female

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 (de Jong & van Joolingen, 1998). 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, etcetera. 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 lay-out 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 Figure 2-1): a guided section with a brief task description and detailed action steps to accomplish the task, and an exploratory section, which offered one or more exercises comparable to the task practiced in the guided section. In line with the minimalist approach (Van der Meij & Carroll, 1995), 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 their 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 (Murphy, Coover, & Owen, 1989) questionnaire was translated into Dutch. This questionnaire used a 5-point agree-disagree scale.

An electronic questionnaire, based on Keller's ARCS theory (Keller, 1983), 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 minutes) in an automatically appearing window (see Figure 2-2). Subjects were asked to

answer the questions by moving the sliders, which always displayed the middle, neutral, position when presented.

Adding interface elements To be able to use the interface, you have to add an interface element to the interface. The interface element for this is available in the Library window. You can find it in the folder: Interface elements. You need a Start button to let the scooter move.

- 1 In the *Library window*, select the folder **Interface Elements**
- 2 Select the subfolder **Dynamic Widgets**
- 3 Select **Action Button**
- 4 Drag **Action Button** from the *Library window* and drop it into the *interface*

Try it yourself As you can see, you cannot stop the car (apart from closing the window). If you want to exercise modifying interfaces more, you can now add a stop-button to the interface. Do not forget to save your work afterwards!

Figure 2-1. Example of a guided section (‘Adding interface elements’) and an exploratory section (‘Try it yourself’) in the manual

The screenshot shows a window titled "Questionnaire" with four questions, each followed by a horizontal slider bar. The questions and their corresponding labels are:

- Question 1: "Is it hard for you to keep concentrated?" with labels "no, not at all" and "yes, very hard".
- Question 2: "Can you apply what you learned, when you are working alone with SimQuest?" with labels "no, definitely not" and "yes, definitely".
- Question 3: "Do you feel confident that you learn the package SimQuest well?" with labels "no, not confident at all" and "yes, very confident".
- Question 4: "Are you satisfied with the product you are developing with SimQuest?" with labels "no, not satisfied at all" and "yes, very satisfied".

At the bottom of the window is an "OK" button.

Figure 2-2. Pop-up motivation questionnaire

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, parts of the exercises that were the same as practiced with the manual,
- items that measured untrained, also known as transfer tasks, new tasks that were different than practiced tasks.

Table 2-2 shows the number of test items in the immediate and delayed test.

Table 2-2. Number of items in Immediate and Delayed test

	Immediate test		Delayed test	
	Trained	Untrained	Trained	Untrained
Number of test items	18	23	19	23

Procedure

The experiment consisted of three sessions: practice, an immediate test, and a delayed test. Before the practice session, subjects answered the questionnaire on gender, age, previous experience with authoring tools and computer experience.

The *practice session* lasted four hours maximally. It was held from 9.00 AM to 1.00 PM with two coffee breaks of fifteen minutes. At the start of this session the subjects were told that their task was to get 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 minutes, 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 PM, and lasted a maximum of two hours. The subjects were asked to try to make the test at their best without the use of a manual. They were also told that some things in the test would be rather different than what they had practiced that morning. They were further told that this session would end at 4.00 PM, 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 two 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 5 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 2-1). Female subjects were scored as 1 and male subjects as 2. Subjects that stated that they had

used an authoring tool at least once before where 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 on (coffee-) breaks was subtracted.

Time used on the guided parts showed a direct effect of manual type on task execution. It shows how long subjects took to complete the reading of the short explanations and carrying 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 a 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 the measurements 2 to 9 were used in the analysis. Examination of the instruments' reliability showed that the questionnaire was highly reliable (see Table 2-3).

Table 2-3. Reliability of motivation measures

	Cronbach's Alpha
Attention	0.95
Relevance	0.96
Confidence	0.94
Satisfaction	0.95
Motivation (combination of the four factors)	0.96

It was tested if the results on the four measures could be combined into one value indicating the subjects motivational state. Table 2-4 shows the correlation 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 differences on motivation between manuals.

Table 2-4. Correlations between motivation measures

	Attention	Relevance	Confidence
Relevance	0.41*		
Confidence	0.58*	0.73*	
Satisfaction	0.47*	0.55*	0.74*

* $p < 0.001$

Coding and scoring - 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 2-5 shows the means and standard deviations of the time users spent on 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 on 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 remains 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%.

Table 2-5. Means (standard deviations) of time in seconds used on guided and exploratory sections

	Manuals		
	V-Part	Text	V-Full
Guided sections	7583 (1071)	7483 (1243)	7427 (1184)
Exploratory sections	596 (597)	1133 (834)*	936 (767)
Total	8179 (1104)	8616 (1169)	8362 (1223)

* $p < 0.05$ compared to V-Part

Motivation

An ANCOVA with computer experience as covariate showed no significant effect of manuals on motivation ($F(2,72) = 0.781$). 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 2-6). These results give an indication of the possible visual relief that this type of manual is supposed to offer when compared to the Textual manual.

Examinations of the results in the course of time showed results that pointed in the same direction, favoring the V-Full manual. Figure 2-3 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 manual. 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.

Table 2-6. Means (standard deviations) of motivational factors (scale 0-100, default score was 50)

	Order of conditions		
	V-Part	Text	V-Full
Attention	64.21 (25.36)	67.59 (20.44)	70.65 (18.84)
Relevance	V-Part	Text	V-Full
	64.31 (22.49)	66.19 (13.57)	69.38 (13.84)
Confidence	Text	V-Part	V-Full
	67.98 (11.58)	68.91 (17.17)	72.23 (10.98)
Satisfaction	Text	V-Part	V-Full
	62.89 (9.95)	63.02 (19.92)	66.86 (11.89)
Motivation	V-Part	Text	V-Full
	65.11 (17.05)	66.16 (11.21)	69.78 (11.94)

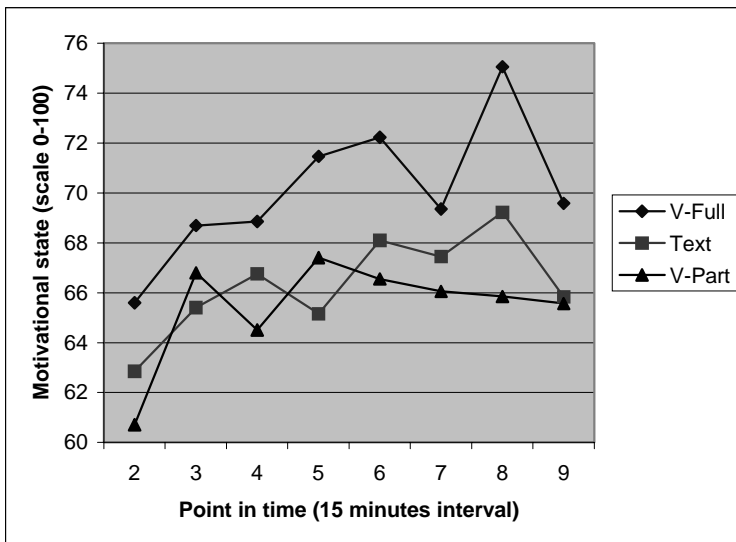


Figure 2-3. Development of motivation in the course of time where high scores indicate high motivation (The neutral (default) score was set at 50.)

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 2-7). 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 2-8). Both the V-Full and the Text manual outperformed the V-Part manual on the delayed test ($(F_{2,71}) = 5,56, p < 0.01$); 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 manual is more difficult. One account may be that the Text manual forces 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 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 that actively constructed their own understanding of the program and therefore disturbed that construction process.

Table 2-7. Means (standard deviations) of test-scores on trained tasks

	V-Part	Text	V-Full
Immediate Test (max. 18)	15.68 (3.92)	15.84 (4.89)	17.13 (1.29)
Delayed Test (max. 19)	16.76 (4.11)	17.32 (2.46)	18.32 (1.09)

Table 2-8. Means (standard deviations) of test-scores on untrained tasks

	V-Part	Text	V-Full
Immediate Test (max. 23)	13.36 (5.44)	14.68 (5.13)	14.87 (4.30)
Delayed Test (max. 23)	11.96 (4.86)	15.40 (4.76)*	16.00 (3.88)*

* $p < 0.05$ compared to V-Part

Trade-off between time and learning effects

Examinations of Pearson correlations between training time and learning effects revealed an intriguing pattern (see Table 2-9). 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 lead to higher test scores respectively, that there was 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 would have been 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.

Table 2-9. Correlations of immediate and delayed test scores with time on guided and exploratory sections

	V-Part	Text	V-Full	Total group
Immediate Test				
Guided sections	-0.59**	-0.41*	-0.15	-0.41***
Exploratory sections	0.16	0.32	0.23	0.26*
Total time	-0.49*	-0.20	-0.00	-0.24*
Delayed Test				
Guided sections	-0.45*	-0.07	0.01	-0.21
Exploratory sections	0.40*	0.31	0.53*	0.42***
Total time	-0.22	0.15	0.35	0.06

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

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 manual are compared, there is no proof that one leads to more learning so far. Also, in time used on guided and exploratory parts, no differences were found between the V-Full and Text manual. 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 to presenting screen captures in manuals is not worth the effort. A closer look at the experiment cautions 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 three 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 4 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 re-start several times, may then have its benefits.

A final consideration is that the subjects that participated in this experiment may not represent computer users in general. As students at 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 regular 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 (1998) 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 that they had trained and even those they did not train, 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 Gellevis (1998) 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 for a visual manual. If the marketing department takes that decision, they should, as this experiment suggests, opt for presenting full rather than partial screen captures.

Acknowledgement

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

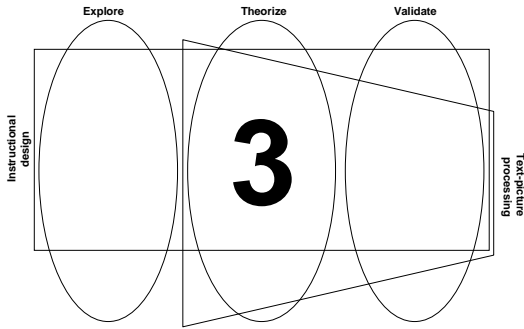
References

- Gardner, D. C., and Beatty, G. J. (1994). *Visuele leermethode Windows 3.1 [Windows 3.1: The visual learning guide]*. Utrecht: A.W. Bruna Uitgevers B.V.
- Horton, W. (1993). Visual Literacy - Dump the dumb screendump. *Technical Communication*, 40, 146-147.
- De Jong, T., and van Joolingen, W. R. (Eds.). (1998). *SimQuest: an authoring system for integrated simulation environments SERVIVE project (ET1020)*. Enschede: University of Twente.
- Keller, J. M. (1983). Motivational Design of Instruction. In C. M. Reigeluth (Ed.), *Instructional-design Theories and Models* (pp. 383-434). Hillsdale, New Jersey: Lawrence Erlbaum Associates.
- Murphy, C. A., Coover, D., and Owen, S. V. (1989). Development and validation of the computer self-efficacy scale. *Educational and Psychological Measurement*, 49, 893-899.
- Price, J., and Korman, H. (1993). *How to communicate technical information*. Redwood City, CA: Benjamin/Cummings.
- Schrifer, K. A. (1997). *Dynamics in document design*. New York: Wiley Computer Publishing.
- Stuur, A. (1996). *Windows voor kinderen: deel2 [Windows for children: part 2]*. Utrecht: A.W. Bruna Informatica.

Van der Meij, H. (1998). Optimizing the joint handling of manual and screen. In J. M. Carroll (Ed.), *Minimalism beyond the Nurnberg Funnel* (pp. 275-310). Cambridge, Massachusetts: MIT press.

Van der Meij, H., and Carroll, J. M. (1995). Principles and heuristics for designing minimalist instruction. *Technical Communication*, 42, 243-251.

Van der Meij, H, and Gellevij, M. R. M. (1998). Screen captures in software documentation. *Technical Communication*, 45, 529-543.



Four Functions of Screen Captures in Software Manuals¹

Abstract - This paper presents a taxonomy of four supportive functions of screen captures in software manuals. It is suggested that screen captures in software documentation can help the user to switch attention, develop a mental model of the program, verify screen states, and identify and locate window elements and objects. Specific research on the use of screen captures in manuals is quite limited and inconclusive. It is argued that a more refined approach, afforded by the taxonomy, is likely to improve practice and research, and can yield evidence supporting the use of screen captures in software documentation.

¹ This text is based on: Van der Meij, H. and Gellevis, M.R.M. (1998). Screen captures in software documentation. *Technical Communication*, 45, 529-543.

Introduction

Screen captures have received little attention in the literature on technical documentation. For example, Brockmann (1990) and Schriver (1997) do not discuss these pictures, although especially Schriver describes many principles of visual design and principles of the interplay between words and pictures. Other handbooks likewise pay little attention to screen captures (Markel, 1994; Price & Korman, 1993; Simpson & Casey, 1988).

Despite this lack of attention, screen captures are considered to be the most frequently used illustration in software manuals (Horton, 1993; Houghton-Alico, 1985). To obtain empirical evidence for this ‘belief’, we randomly took one hundred software manuals and, again randomly, selected a single page from each manual showing one or more pictures. We found that:

- Seventy-six percent of the pages showed one or more screen captures. Nearly all screen captures were pictures from the whole screen, or pictures from one or two windows of the program. Only six of the over one hundred pictures showed a single object like a button or icon.
- Twenty-three percent of the pages had a picture in the form of a schema, flow chart, diagram, or table.
- Fifteen percent of the pages showed an icon or a symbol that identified an information type such as a note, tip, or warning.
- Six percent of the pages showed an icon or a symbol of the manipulation device (usually the mouse).

These findings show that screen captures are indeed the most frequently used type of picture in software manuals. The claimed reason for this dominance is that screen captures can help the user to acquire the necessary knowledge and skills because they can convey some things better than other illustrations or words (Horton, 1993), but there is little empirical research to substantiate this claim. In addition, the claim gives no real guidance for creating and applying effective screen captures. This paper presents a taxonomy on functions of screen captures in software manuals. Figure 3-1 summarizes these functions.

Functions <i>Screen captures can help the user to:</i>
- switch attention
- develop a mental model of the program
- verify screen states
- identify & locate window elements and objects

Figure 3-1. Four main functions of screen captures in software manuals

The four functions were derived from an analysis of the user's task while working with the software manual and computer simultaneously. Figure 3-2 shows a typical part of a software manual. Analyzing this manual-part with its accompanying user behavior leads to the following user tasks.

First, the manual intends to get the user to interact with the system. This is evoked by the action steps 1 to 6, which prompt the user to do something with or on the computer. The action steps require the user to *switch attention* between the manual and the computer screen.

Second, the instructions and explanations in general are meant to teach the user about, and how to work with the computer program. The ultimate goal is that the user can create things and solve problems with the computer program. In this case, it teaches how to change or create an interface, explains what tools can be used for this, and what these tools entail. These tasks contribute to the *construction of a mental model* of the program.

Third, the manual tries to make sure that users stay on the right track by building in check-points and giving information to overcome possible errors. Action step 6 is an example in which the user is asked to *verify the screen state* with the information in the manual.

Fourth and finally, the manual tries give support by informing the user about the objects they have to act upon, and where to find these. An example of such a specific explanation of buttons or objects is given in the last part of the example in Figure 3-2, where the buttons of the Canvas tool are explained to help users *identify* the objects. Support for identification is also given in action step 2. This step not only prompts the user to do something, it also explains, what a slider is used for, namely scrolling through a list. Apart from these explanations, throughout the whole example, pointers are given where to find a particular object, usually printed in italics to support *locating* these objects.

The presentation of screen captures in a manual can take various forms. We suggest four dimensions on which the design of a screen capture can vary. These dimensions are: coverage, position, size, and cueing. In Appendix D, these dimensions are explained and illustrated. Although a function and design are related, in this book we mainly focus on the functions. The particular design used for a function is therefore the expected best fit between the function of a screen capture and its design, based on the design dimensions.

The next sections discuss how screen captures can support these four functions or tasks. For clarity of presentation it is ignored that a screen capture can serve more than one function. Each section first discusses the impact of each function on the user and offers arguments about the benefits of using screen captures for support. Then, examples from existing manuals are presented to illustrate the various expressions or design solutions for each function including designs based on the example-manual presented in Figure 3-2. The sections conclude with findings from empirical research about the concerned function.

Changing an interface

- 1 Click on the *tabsheet level 1*
- 2 In the *Application: motion* window, use the **slider** to scroll down in the list of elements
- 3 In the folder *Interface Elements*, click on **1 Velocity scooter**
- 4 Click with your **right** mouse button
- 5 In the *pop-up menu*, click on **Edit**
- 6 Check, in the *Edit: 1 Velocity scooter* window, if the velocity meter is placed directly under the start button

If the velocity meter is not directly placed under the start button, replace it by clicking on the velocity meter, holding the left mouse button down, and dragging it to the appropriate place

You can lay-out your interface using the buttons of the Canvas Tool.
The functions of these buttons are:

Alignment

- Top side (*1st button*)
- Center horizontally (*2nd button*)
- Bottom side (*3rd button*)
- Left side (*4th button*)
- Center vertically (*5th button*)
- Right side (*6th button*)

Interspace

- Horizontal space equal from center (*7th button*)
- Horizontal space equal from sides (*8th button*)
- Vertical interspace equal from center (*9th button*)
- Vertical interspace equal from sides (*10th button*)

Height and width

- Equal height (*11th button*)
- Equal width (*12th button*)

Figure 3-2. Typical example of a part of a (textual) software manual

Switching attention

The user of a manual has to carry out a complex task because he or she must deal with several information sources (manual and computer screen) and input devices (keyboard and mouse) simultaneously. An important difficulty in handling these sources is dealing with their interrelationships. Users must regularly switch attention to and from input device, manual, and screen.

Especially novices may easily make mistakes caused by such coordination problems. A typical problem of this kind may occur when the user reads an instruction, carries it out, and then performs the next step, “never checking to see if the step worked properly, or if anything happened at all” (Carroll, 1984, p. 126). This nose-in-the-book syndrome is a well-known example of a user failing to switch attention when needed. Contrary to this first risk, current graphical interfaces may be

so attractive and can present so much information, that users may start exploring the system without following the instructions in the manual and that way easily lose track.

Screen captures can help the user with switching attention by: (1) prompting the user to attend to the screen at the right moment, and (2) providing a clear point for re-entry into the manual after attending to the screen. These processes can be clarified by inserting screen captures in the (textual) action steps presented in Figure 3-2. The screen capture in Figure 3-3 prompts users to switch attention to the screen regularly. Because the screen capture is not clearly readable, it stimulates the user to look up to the screen for each instruction. Figure 3-4, taken from a commercial manual, illustrates the same function. It cues the user that it is time to look up to the screen. The two screen captures (with some text) present coordinative information, prompting the user to attend to the screen. The presence of a screen capture after each action step also makes it easy for the user to get back to the right place in the manual after having looked up at the screen.

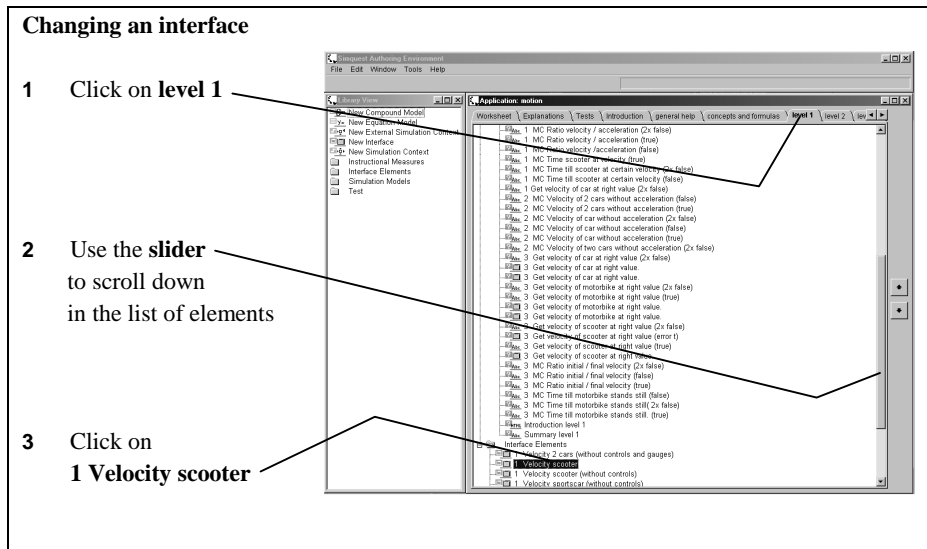


Figure 3-3. Screen captures in which objects are visibly connected to user instructions support frequent switches to and from the computer screen

No research has yet been reported which shows that users benefit from screen captures for switching behavior. There are, however, other features of a manual that may stimulate attention switching. Research on minimal manuals has shown that users sometimes carry out as much as 90% of the instructions (Van der Meij, 1997), which shows that users indeed follow a great deal of the prompts given in the

manual. It can therefore be expected that these prompts supported by screen captures may even become stronger, to get users even more easily into a ‘switching mode’.

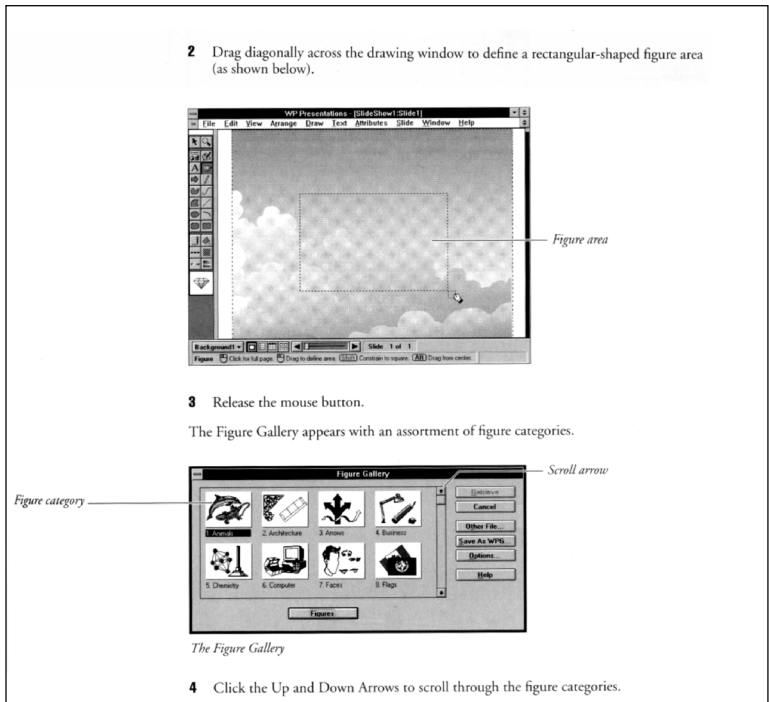


Figure 3-4. Screen captures can prompt users to switch attention to the screen at the right moment, and provide a clear point for re-entry into the manual afterwards

Developing a mental model of the program

A mental model is a user’s understanding of how a program works. It helps the user construct actions and explain why these actions produce particular results (Carroll & Olson, 1988). A mental model has to do with the structure of the program. Users build this model while they become used to how the program works and what it looks like.

Mental models play a critical role in problem solving. Users apply their mental model for detecting, defining, diagnosing, and solving problems. A strong mental model makes it possible for users to deal with new situations, it is conditional for handling transfer tasks. Experience, training, and imitation all contribute to mental model development. They lead to “...a model that evolves in the mind of a user as he or she learns and interacts with a computer system. ... It represents the structure and internal relationships of a system ... and is the source of the user’s expectations about the effects of actions; it can guide navigation or planning of actions and contribute to

interpretation of feedback” (Jih & Reeves, 1992, p. 45)². Screen captures can thus contribute to the development of a mental model by: (1) acquainting the user with the main windows, (2) explaining the spatial layout of a window, and (3) developing a sense of logical flow, or progression, of windows.

To acquaint the novice user with the main types and ensure active processing of each window, these windows can be presented in the manual while the user performs basic tasks. A screen capture that appears during task execution enforces the user to study the window. The investment of mental effort is vital for the development of a mental model (Peeck, 1993; Winn, 1993). In Figure 3-5, the user becomes acquainted with a few important windows in the course of carrying out a task. The example is part of a procedure for saving a file. The screen captures acquaint the user with the main windows as they appear in sequence during task execution. Windows and the pop-up menu are displayed in their correct proportions to convey the real look of the program.

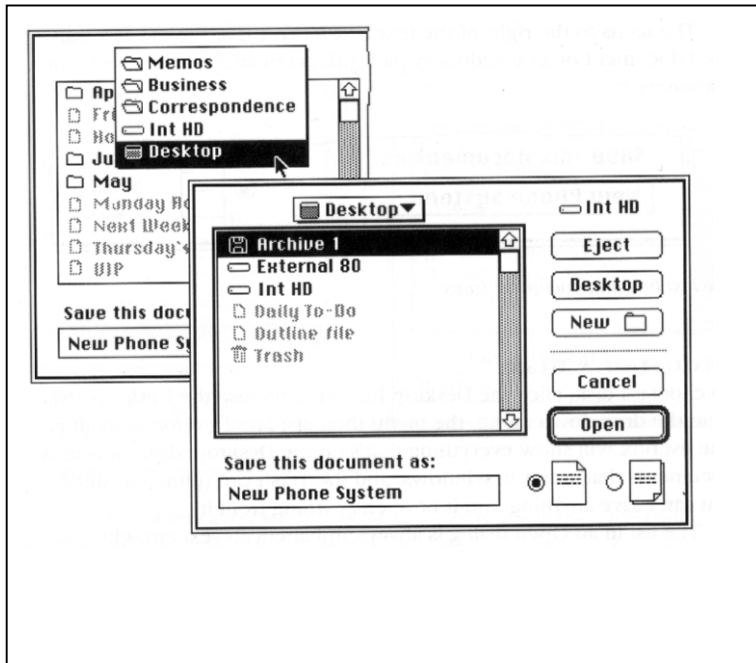


Figure 3-5. To acquaint novices with the main types of windows of a program it is beneficial to present these in the course of doing real work

² Jih and Reeves derived this definition from Gentner & Stevens, 1983; Norman, 1983; and Van der Veer, 1989.

Users must develop some understanding of the design or structure of successive screens. This may come about through repeated exposure to a program, but it is probably more effective to support this learning process by explaining the spatial lay-out of a screen. Figure 3-6 shows a screen capture that is part of a procedure that instructs users to create slides. Leader lines and captions give a pointed description of the structural elements of the window.

The presentation of a series of screen captures that show the progression of windows during task execution can also be a strong aid for developing a mental model. This is illustrated in Figure 3-7, which is the follow-up of Figure 3-3, based on the example manual in Figure 3-2.

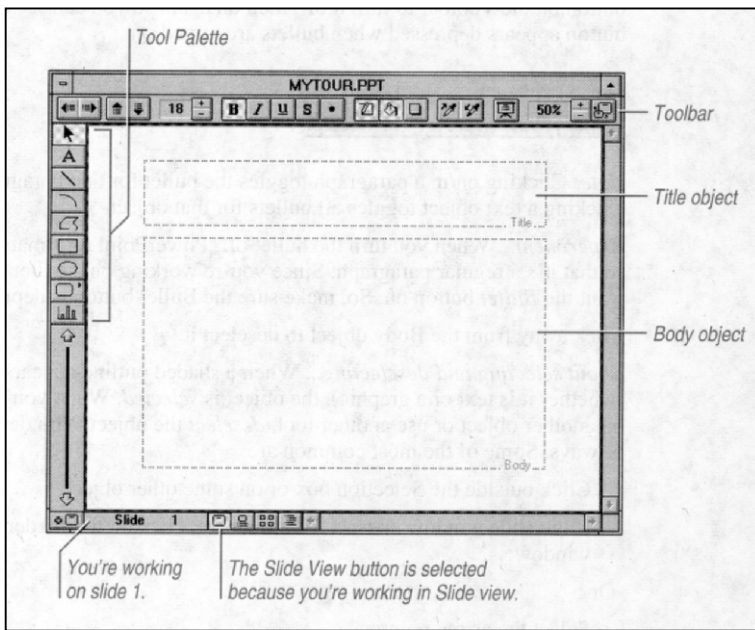


Figure 3-6. Explaining the lay-out of a window contributes to mental model development

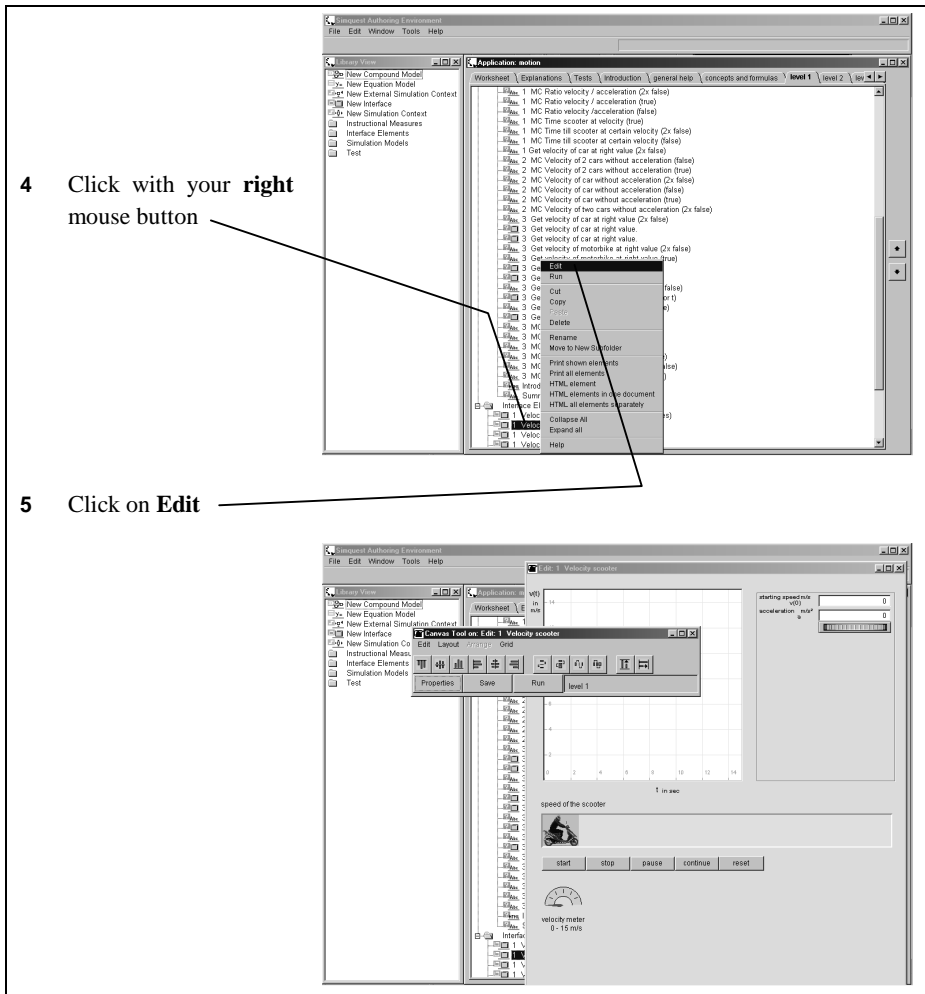


Figure 3-7. The presentation of a series of screen captures that catch the progression of windows can strongly affect mental model development

It is difficult to find a good way to directly measure the development of a user's mental model. The following definition for measuring a mental model in research on user documentation is suggested. A mental model is the knowledge and skills used to solve trained and untrained problems without support from a manual after instruction.

Empirical research indicates that screen captures have a significant effect on the development of a user's mental model. Two studies, found that users' mental models differed depending on the kind of screen captures they encountered during instruction (Gellevej, Van der Meij, De Jong, & Pieters, 1999; Van der Meij &

Jenné, 1999). Subjects who used a visual manual that consisted only of full screen captures developed a significantly better mental model than did subjects who used a visual manual with predominantly partial screen captures. When compared to a textual manual, however, no difference was found between a textual and full screens manual (Gellevij et al., 1999).

Verifying screen states

Many people still see a computer as a delicate instrument that can easily be broken. Especially novice users may fear that their actions will damage the computer or the application. Their attitude towards working with computers may therefore initially be rather reserved. Research corroborates this impression. As people become more experienced, their attitude towards the computer changes and their self-confidence increases (Gardner, Dukes, & Discenza, 1993; Pope-Davis & Twing, 1991). The manual can alleviate initial fears by presenting information that allows users to verify that they are on the right track. For example, after pressing a key, novices often want to know whether they have done the right thing. A potentially strong form to present such verification information is screen captures. Screen captures make it possible to verify screen states and, thus, confirm or contradict the user's progress (Horton, 1993; Price & Korman, 1993).

Apart from providing positive feedback, screen captures can help users detect errors early on and thus facilitate error management. Screen captures are supposed to be optimally suited for this verification process. They make it relatively easy to compare the information in the manual with the actual computer screen. In contrast to using text, the user does not need to process or code the textual information into a visual; comparing a picture of the screen with the actual screen is easier than comparing the content of written text with a screen.

The contribution of screen captures to verification is not restricted to learning situations. Screen captures can also be applied in cases where documentation is used for referential purposes. That is, when users want to "get in, grab the relevant information, and get out and back to their work as quickly as possible" (Redish, 1998, p. 223). There are two situations in which screen captures can help users verify their actions such a random access approach to documentation.

The first situation is when the user needs the manual after a warning or an error message on the screen. An inventory of sixty manuals shows that in 86% of the manuals the user is required to read the complete page because there is no form of highlighting that points to the relevant information. A screen capture can serve as such a cue, simplifying the verification process.

Another situation when users benefit from screen captures for verification is right before the start of a procedure to verify whether the program is in the right mode. The user can check such conditions based on textual information, but a combination of text and picture more strongly attracts the user's attention and simplifies verification.

Screen captures can thus help users with screen verification by: (1) supporting progress checks, and (2) facilitating (re)entry into the manual.

Screen captures presented at critical moments in a procedure mark the user's progress. Figure 3-8 exemplifies this. The example describes the procedure of creating a chart. The screen captures make it easy for users to verify whether they have done the right thing by pointing to the exact parts of the windows to be filled in or modified.

Figure 3-9 illustrates a situation in which a screen capture is an easy point of entry for a user who browses the manual for help on a particular problem.. The example explains alert messages. The screen capture catches the user's eye and supports verification. A quick scan can reveal whether the screen capture is what the user is looking for. The cued text wrapped around the picture invites further reading.

Another example of a screen capture designed to support verification is shown in Figure 3-10. This example is again based on Figure 3-2. Only the relevant window is shown (and not the full screen) to provide for readability. In addition a cueing technique is used to point the user to the exact part to be verified.

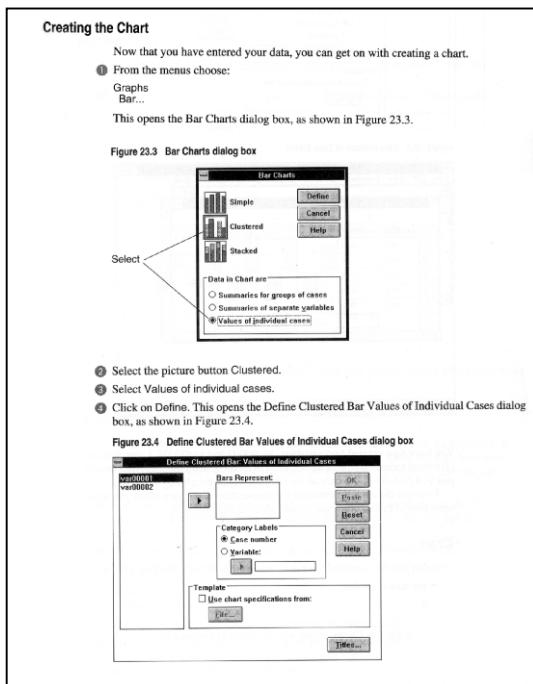


Figure 3-8. Screen captures presented at critical moments in a procedure support verification by marking the user's progress

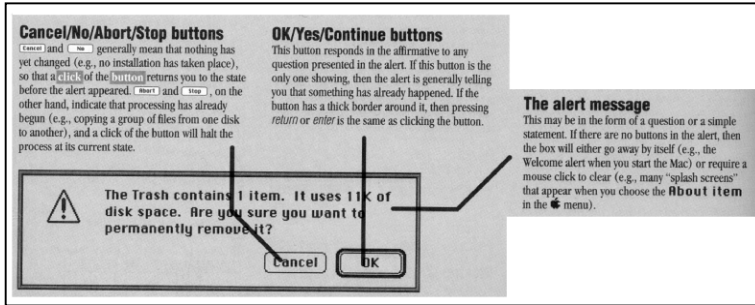


Figure 3-9. A screen capture can be an easy point of entry for random access use of a manual

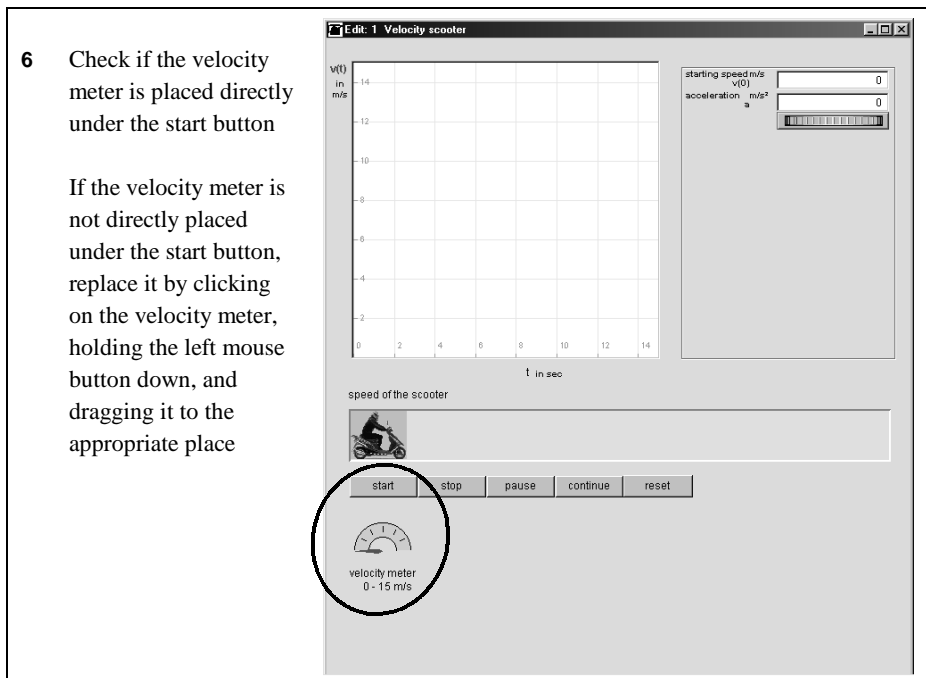


Figure 3-10. A screen capture to support verification

Practitioners state that one of the main advantages of the display of screen captures is their support for verification (Horton, 1993; Price, 1984). For example, Price advises to include numerous displays in tutorials to let users know that they have done the right thing and gotten the right display. Practitioners also assert that these displays do not merely help users verify that they are still on the right track, but that they

simultaneously contribute to user motivation. They have a reassuring effect and they help build confidence.

Research has not explicitly studied the contribution of screen captures on verification, but its motivational counterpart has been examined. Empirical studies have not shown that the presence of screen captures increases user motivation (Gellevij et al. 1999; Van der Meij & Jenné, 1999).

Identifying and locating window elements and objects

The windows in contemporary software programs are largely filled with objects that can be manipulated in various ways. For example, the startscreen of Microsoft Word 6 for the Macintosh displays sixty-three objects (menu-options, icons, and symbols) that can be clicked, or moved. Their logical organization and the use of icons and symbols reduces, but does not abolish the need for explanation. This need for explanation was shown by Van der Meij (1995), who asked university students who were moderately experienced users of Microsoft Word to write down the meaning of the icons and symbols of Word's ribbon and ruler. The students gave a correct answer for only 48% of cases. They also frequently failed to infer the meaning of an icon or symbol. For example, knowledge of the meaning of the left alignment symbol was no guarantee that students also could infer the meaning of the other alignment symbols.

An important contribution of screen captures is reduction of task complexity (Sweller & Chandler, 1994; Van der Meij, 1998). Screen captures can reduce cognitive load by making it easier for the user to identify and locate the object(s) needed for task execution. A screen capture can help guide the user's attention on the relevant part of the window and in that way simplify locating that part. Screen captures are also helpful when there is a risk of confusion by explaining the elements in a window, supporting identification of the element. In these instances screen captures can speed up task execution and reduce the possibility of making errors.

Associated with the large amount of objects offered in program windows, is that many program options are chronically underutilized. For example, one study found that 170 users of Unix used no more than 20 of the available 400 commands for about 70% of their time (Kraut, Hanson, & Farber, 1984). Another study found "no evidence that more powerful commands were being used by experienced users" (Rosson, 1984, p. 174). Both studies are from the early eighties, but more recent research gives comparable results (e.g., Bhavnani & John, 1997). Screen captures can contribute to alleviating novelty and efficiency aspects of underutilization. They can make users aware of and motivate them to use untried task options. In a similar way, screen captures can motivate users into moving from sufficient into efficient usage of a program. They can, for example, prompt users to use short cuts by selecting an icon rather than a series of menu choices.

In short, screen captures can help reduce task complexity and alleviate the problem of underutilization by identifying and locating window elements or objects for the user. Screen captures can do this by: (a) identifying window elements or objects, and (b) locating window elements or objects.

Figure 3-11 shows support for identification in which the screen capture is integrated with the text, and offered during task execution. Because information about the location is not given, the user is stimulated to process the information on the screen more deeply (Van der Meij, 1996a, 1998). Figure 3-12 also illustrates identification of window elements and objects. Based on combination of text and picture the objects in the Tools menu of HyperCard are identified. Note that each icon is displayed out of context. Each object can and must be studied on its own.

Users who search for a particular icon or button without knowing its appearance or place, benefit from screen captures that identify and locate such an object. Figure 3-13 shows an example. The pictures are performance-support aids and for this reason they appear on a tear-off card in the back of the manual. The screen captures describe and position each object. In addition, they give an overview in which users can easily make comparisons between objects. Figure 3-14 shows a comparable design for the example manual of Figure 3-2.

Apart from the support given for locating by screen captures as shown in Figure 3-14, locating is also supported in the previous three figures that were based on the manual example in Figure 3-2. Figure 3-3 and Figure 3-7 make locating easy by the hairlines that point to the relevant part of the screen. In Figure 3-10, this is realized by the circle around start button and velocity meter. In all cases, the textual help on locating (printed in italics in Figure 3-2) becomes redundant and can therefore be omitted.

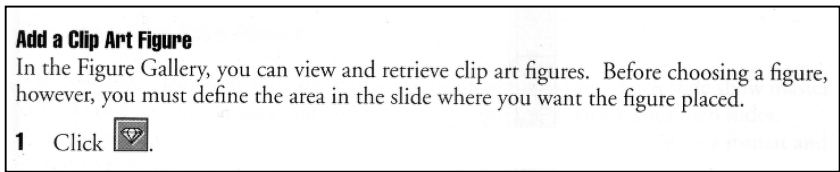


Figure 3-11. A screen capture that forms an integral part of an instruction can hardly be ignored and identifies what the user should see on the window

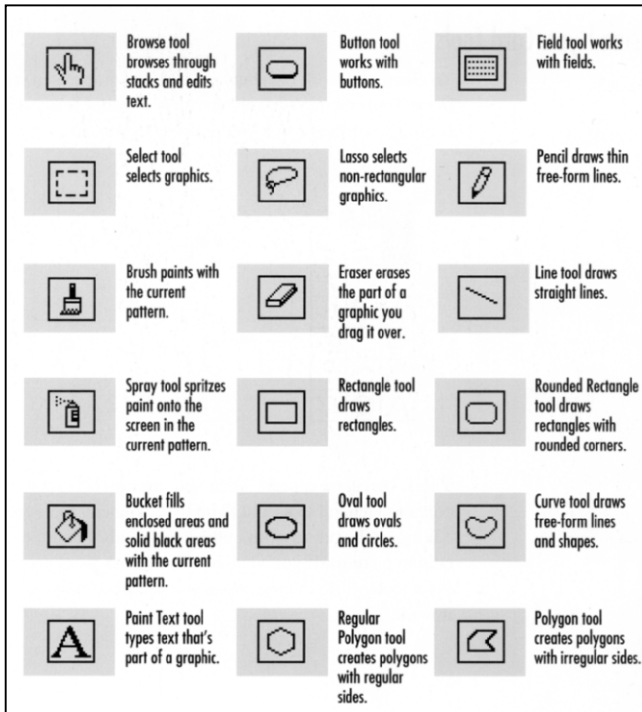


Figure 3-12. A screen capture can support object identification by a joint presentation of objects, out of their location

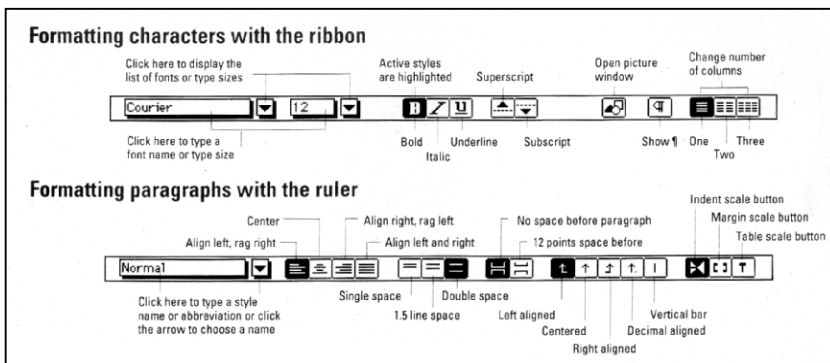


Figure 3-13. Unlike some forms of on-line help, a screen capture can offer an overview to support object identification and location

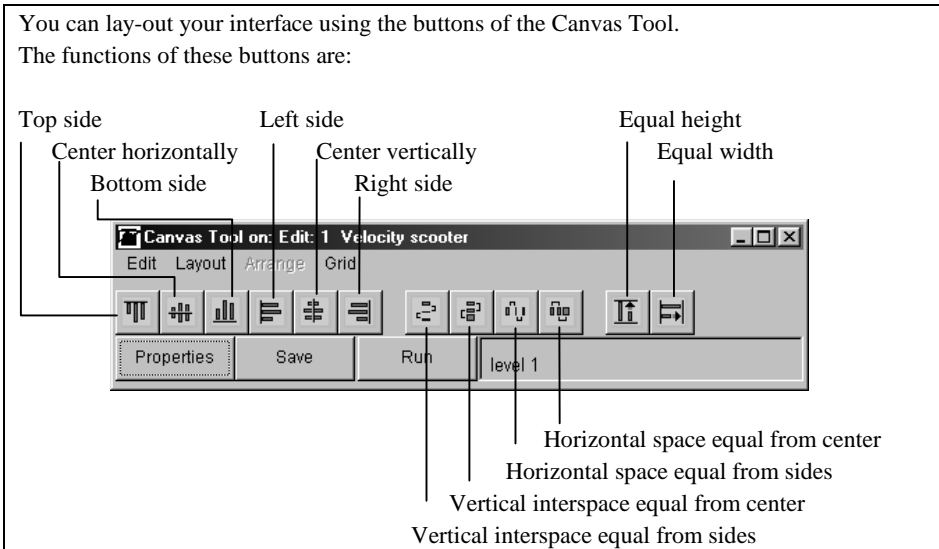


Figure 3-14. A screen capture to support identification and locating of screen elements and objects

Research has not studied the processes of identifying and locating objects directly, but there is indirect proof that screen captures can have a substantial effect on the locating process. This proof comes from a study in which users were trained to use a new database program (Van der Meij, 1996a). In the study, one group of subjects trained with a manual with full screen captures while another group used a manual without screen captures. While quality of task execution was equal for both groups, the screen captures sped up the subjects' task completion with about 35%. This effect can be attributed to the pictorial support given for locating objects.

Conclusion

This paper discusses four functions of screen captures in software manuals. It is argued that screen captures should not be used as illustrations that can make a manual more visually appealing, but as specific meaningful forms of instructional support. Using screen captures from such a function-oriented perspective is expected improve task execution and learning.

Earlier research on screen captures cannot easily be linked to the four functions because the screen captures used in these experiments were not deployed according to these functions. Studies in which the screen captures are employed and designed for their specific function in the user's task may provide for a clearer and more consistent view on the benefits for screen captures in software manuals.

References

- Bhavnani, S. K., & John, B. E. (1997). From sufficient to efficient usage: An analysis of strategic knowledge. In S. Pemberton (Ed.), *Looking to the future: CHI'97 conference proceedings*. New York, NY: Association for Computing Machinery.
- Bradford, A. (1988). Learning to write with a word processor. In R. Krull (Ed.), *Word processing for technical writers* (pp. 160-169). Amityville, NY: Baywood.
- Brockmann, R. J. (1990). *Writing better user documentation: From paper to hypertext (version 2.0)*. New York, NY: John Wiley and Sons.
- Carroll, J. M. (1984). Designing minimalist training materials. *Datamation*, 30, 125-136.
- Carroll, J. M., & Olson J. R. (1988). Mental models in human-computer interaction: Research issues about what the user of software knows. In M. Helander (Ed.), *Handbook of Human-Computer Interaction* (pp. 45-65). Amsterdam: Elsevier.
- Gardner, D. G., Dukes, R. L., & Discenza, R. (1993). Computer use, self-confidence, and attitudes: A causal analysis. *Computers in Human Behavior*, 9, 427-440.
- Gellevij, M., Van der Meij, H., De Jong, T., & Pieters, J. (1999). The effects of screen captures in manuals: Textual and visual manuals compared. *IEEE Transactions on Professional Communication*, 42, 77 - 91.
- Horton, W. (1993). Dump the dumb screen dumps. *Technical Communication*, 40, 146-148.
- Houghton-Alico, D. (1985). *Creating computer software user guides*. New York, NY: McGraw-Hill.
- Jih, H.J., & Reeves, C. T. (1992). Mental models: A research focus on interactive learning systems. *Educational Technology Research and Development*, 40, 39-54.
- Kraut, R. E., Hanson, S. J., & Farber, J. M. (1984). Command use and interface design. In A. Janda (Ed.), *Human Factors in Computing Systems. Proceedings of the 1983 CHI Conference on Human Factors in Computing* (pp. 120-124). North Holland: Elsevier.
- Markel, M. (1994). *Writing in the technical fields. A step by step guide for engineers, scientists, and technicians*. Piscataway, NJ: IEEE Press.

Peeck, J. (1993). Increasing picture effects in learning from illustrated text. *Learning and Instruction*, 3, 227-238.

Pope-Davis, D. B., & Twing, J. S. (1991). The effects of age, gender, and experience on measures of attitude regarding computers. *Computers in Human Behavior*, 7, 333-339.

Price, J. (1984). *How to write a computer manual: A handbook of software documentation*. Menlo Park, CA: Benjamin/Cummings.

Price, J., & Korman, H. (1993). *How to communicate technical information*. Redwood City, CA: Benjamin/Cummings.

Redish, J. (1998). Minimalism in technical communication: Some issues to consider. In J. M. Carroll (Ed.), *Minimalism beyond the Nurnberg funnel* (pp. 219-245). Cambridge, MA: MIT Press.

Rosson, M. B. (1984). Patterns of expertise in text editing. In A. Janda (Ed.), *Human Factors in Computing Systems. Proceedings of the 1983 CHI Conference on Human Factors in Computing* (pp. 171-175). North Holland: Elsevier.

Schrivers, K. A. (1997). *Dynamics in document design*. New York, NY: Wiley.

Simpson, H., & Casey, S. M. (1988). *Developing effective user documentation*. New York, NY: McGraw-Hill.

Sweller, J., & Chandler, P. (1994). Why some material is difficult to learn. *Cognition and Instruction*, 12, 185-233.

Van der Meij, H. (1995). Snap u de pictogrammen? [Do you understand the icons?] *Tekst[blad] 1*, 59-60.

Van der Meij, H. (1996a). A closer look at visual manuals. *Journal of Technical Writing & Communication*, 26, 371-383.

Van der Meij, H. (1996b). "Does the manual help? An examination of the problem-solving help offered by manuals". *IEEE Transactions on Professional Communication*, 39, 146-156.

Van der Meij, H. (1997). The ISTE approach to usability testing. *IEEE Transactions on Professional Communication*, 40, 209-223.

Van der Meij, H. (1998). Optimizing the joint handling of manual and screen. In J. M. Carroll (Ed.), *Minimalism beyond the Nurnberg funnel* (pp. 275-309). Cambridge, MA: MIT Press.

Van der Meij, H., & Jenné, M. A. (1999). *Designing and testing visual manuals for the adult learner* (Report for the American Society for Technical Communication (STC)). Enschede, The Netherlands: Twente University, Faculty of Educational Science and Technology.

Winn, W. (1993). Perception principles. In M. Fleming, & W.H. Levie (Eds.), *Instructional message design. Principles from the behavioral and cognitive sciences (2nd edition)* (pp. 55-126). Englewood Cliffs, NJ: Educational Technology Publications.

Figure references

3-4. WordPerfect Corporation. (1993). *WordPerfect Presentations Version 2.0 Reference*. Orem, Utah: WordPerfect Corporation, pp. 46-47.

3-5. Aker, S. Z. (1991). *The Macintosh Companion: The basics and beyond*. Reading, MA: Addison-Wesley Publishing Company Inc., pp. 250.

3-6. Microsoft Corporation. (1992). *Microsoft powerPoint Handbook*. Microsoft Corporation: Ireland, pp. 33.

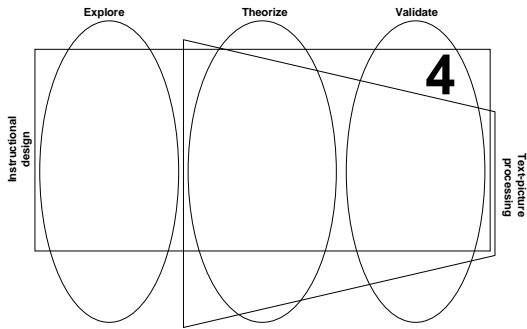
3-8. Norusis, M. J. (1993). *SPSS for Windows Base System User's Guide Release 6.0*. Chicago, IL: SPSS Inc., pp. 441-443.

3-9. Goodman, D. (1992). *Danny Goodman's Macintosh Handbook*. New York, NY: Bantam Books, pp. 104.

3-11. WordPerfect Corporation. (1993). *WordPerfect Presentations Version 2.0 Reference*. Orem, UT: WordPerfect Corporation, pp. 46.

3-12. Apple Computer Inc. (1993). *Hypercard Reference Manual*. Cupertino, CA: Apple Computer Inc., pp. 4-10.

3-13. Heid, J. (1992). *Macworld Guide to Microsoft Word 5*. San Mateo, CA: IDG Books Worldwide Inc.



Validation: Switching Attention¹

Abstract – This study set out to validate the supportive role of screen captures for switching attention. Forty-two participants learned how to work with Microsoft Excel with a paper manual. There were three types of manuals: a textual manual, a visual manual with full screen captures and a visual manual with a mixture of partial and full screen captures. The findings show that participants in all conditions looked up from the manual to the screen on about 97% of the cases in which such a switch was called for. Rank order analyses showed that users of the visual manuals switched attention significantly more often than did users of the textual manual. No differences were found between conditions on learning effects and training time.

¹ Gellevij, M.R.M., & Van der Meij, H. (2002). Screen Captures to Support Switching Attention. *IEEE Transactions on Professional Communication*, 45.

Introduction

Users who consult a manual to learn how to use a computer program run the risk of falling into the nose-in-the-book syndrome (Carroll, 1984). That is, they may keep their attention too much focused on processing the manual at the expense of paying too little attention to what happens on the computer screen. Contrary to this first risk, current graphical interfaces may be so attractive and can present so much information, that users may start exploring the system without following the instructions in the manual and that way easily loose track. To remedy these risks, the manual can stimulate users to look back and forth from the manual to the screen regularly². This study is set out to examine the function of screen captures that serves that switching purpose.

Screen captures seem well-suited to support switching behavior. They simplify the process of applying the information presented in the manual. In a purely textual manual the user needs to put in cognitive effort in finding the right match between the textual description and the corresponding pictorial representation on the screen. Screen captures circumvent this problem. With screen captures the user does not need to connect divergent modalities; the user must match identical representation modes. At the same time, screen captures simplify the process of getting back from the computer screen to the right place in the manual. Screen captures offer better support for re-entry than do written statements because they stand out more on the page. They facilitate access into the manual because they are easier to perceive.

To date, no research has yet been reported which shows that users benefit from screen captures for switching behavior. There are, however, other features of a manual that may stimulate attention switching. Research on minimal manuals has shown that users sometimes carry out as much as 90% of the instructions (Van der Meij, 1997), which shows a great deal of compliance with the action-oriented focus of this manual design. The frequency of attention switching is likely to be high as well because minimal manuals, along with many recent manuals, tend to get users more easily into what is perhaps best characterized as a 'switching mode'. When instructions to act prevail over the presence of conceptual information, the manual is more likely to facilitate the development of a switching habit. Still, even in such manuals, screen captures may strengthen switching attention behavior, although the potential benefits then presumably are relatively small.

² The argument that problems with attention switching can be solved by using online help is only partly true as online help too involves switching. Indeed users sometimes make mistakes in thinking that the help is the application itself. In addition, a change from paper to online help support introduces other drawbacks such as the difficulty of reading from a computer screen, and the difficulty of handling both the source and the help information in a limited screen with overlapping windows.

The goal of this study is to examine the influence of screen captures on switching behavior in an action-oriented manual. The effects of two types of visual manuals and a textual (control) manual were studied. One visual manual presented full screen captures (Visual-Full) and the other contained a mixture of partial and full screen captures (Visual-Part&Full). Figure 4-1 and Figure 4-2 illustrate the two manual types.

The difference between the two visual manuals lies in the representation of the object that the user must act upon. In the Visual-Full manual the user must examine the screen capture and computer screen to find that object. In the Visual-Part&Full manual the object is specifically shown in the partial screen capture. Finding the object becomes easier that way but at the same time makes the Visual-Part&Full manual more self-contained

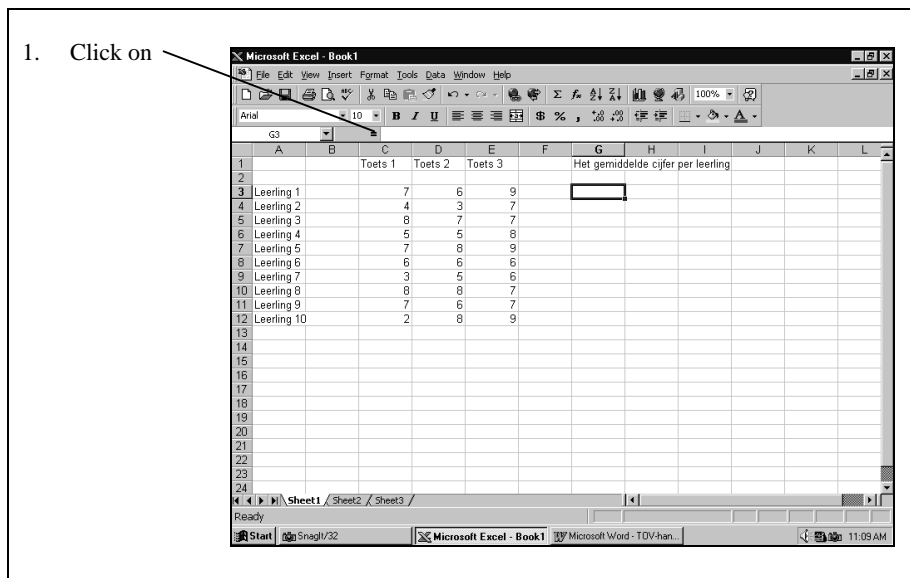


Figure 4-1. Example of an instruction in the Visual-Full manual

When considering the advantages and drawbacks of the two visual variants on the support they can give to switching attention, it is difficult to anticipate their effects. There are three aspects of the screen capture designs that determine such effects: readability of the screen captures, self-containedness of the manual, and redundancy of screen captures.

Regarding the readability of the screen captures, the Visual-Part&Full manual seems the best design to support switching attention. After all, the object that needs to be clicked on is presented in a readable format and connected to the full screen capture, which makes finding that object on the computer screen relatively easy.

Easier than when this partial screen capture is omitted, which is the case in the Visual-Full manual. Here, only the ‘hard to read’ full screen is presented, which may make it more difficult to find the object. It can be expected that the ease with which the relevant object on the screen can be found, promotes switching behavior. This way, the Visual-Part&Full manual seems the best design for switching attention.

Regarding self-containedness of a manual, the Visual-Full manual seems the best design to support switching attention. The action step, connected to the full screen capture that is difficult to read, forces the user to look up to screen to find the relevant object. In contrast, because of the combination of the partial and full screen captures, in the Visual-Part&Full manual it is possible to read the instructions without carrying them out. This self-contained Visual-Part&Full manual most probably does not stimulate switching as well as a manual that is less self-contained.

Regarding redundancy of screen captures, the Visual-Full manual seems the best design to support switching attention. The partial screen capture in combination with the full one in the Visual-Part&Full manual present the same information twice. Both need to be processed in working memory, which may increase cognitive load (Sweller & Chandler, 1994). Because of that increase, switching behavior may be hindered.

The Visual-Full manual seems to be the best screen capture design for switching attention on two out of the three aspects discussed above. To predict that this design will therefore be the better of the two is rather simplistic, as it hard to envision that all three aspects have an equal impact on the potential effects. Therefore, no prediction is given about which of the two visual manuals best supports switching.

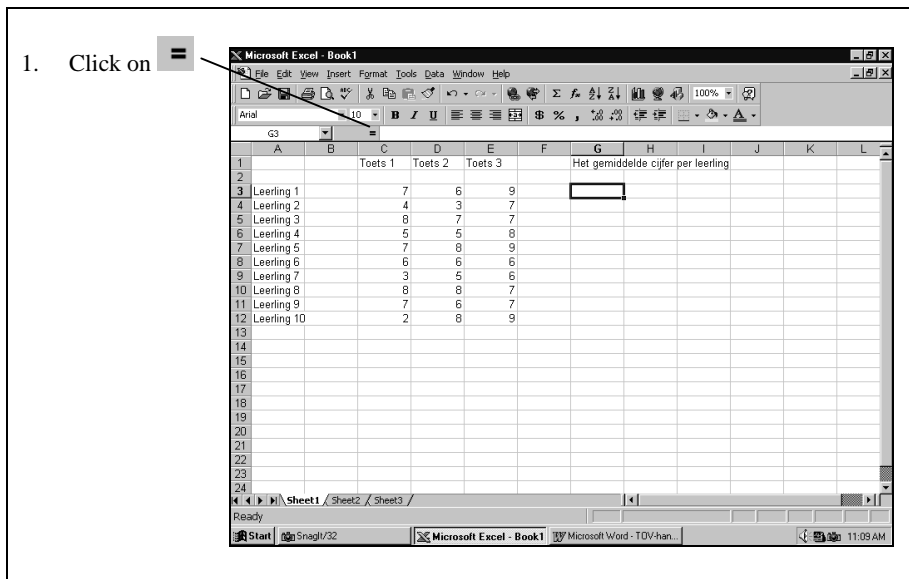


Figure 4-2. Example of an instruction in the Visual-Part&Full manual

A textual manual was used in the control condition. The visual manuals were expected to stimulate users to switch attention more often than the textual manual because the screen captures used in the visual manuals simplify the processes of looking up from the manual to the screen and from the screen back into the manual, as argued earlier.

In addition to switching behavior we also examined training time, and learning effects on a post test. Previous research has shown that visual manuals and textual manuals can have very different effects on these variables. However, compared to those studies, the training task in the present study is relatively short, which may make it difficult to find such differences.

Method

Participants

Forty-two students (6 males and 36 females) from the Faculty of Educational Science and Technology from Twente University participated in the study. The computer knowledge and skills of this group, measured by self-grading on a scale from one to 10, was moderate to high with a mean score of 6.90 (SD = 0.88). Participants had no experience with Microsoft Excel, the topic of the manuals. They participated on a voluntary basis.

Materials

Manuals

The designs of the manuals were based on the findings from previous studies on screen captures. A Guided Tour design (Gardner & Beatty, 1994) was used as the primary basis for the two visual manuals. Gellevij, Van der Meij, De Jong, and Pieters (1999) have shown that this manual which displays only full screen captures, leads to more learning compared to a manual that uses only partial screen captures. In the Guided Tour approach, written instructions are presented on the right hand side of the page with the full screen capture presented on the left. Van der Meij (2000) found that learning improves with a reversed order. By presenting the instructions on the left hand side of the page and the screen captures on the right, the design preserves the reading direction of western audiences and thus better supports the processing of the information in the manual.

The Visual-Full manual is presented in Figure 4-1, the Visual-Part&Full manual is shown in Figure 4-2. In both manuals the instruction is connected with the full screen capture by a thin hairline. More importantly, the written instruction is incomplete without the picture. The user must attend to the screen capture to find the object that must be acted on.

The third manual contained no screen captures. To compensate for the absence of visual cues for identifying and locating screen objects, such information was added to the verbalized instructions in this textual manual. That is, objects that would be depicted in the instructions in the Visual-Part&Full manual were described or typed, along with information about their location on the screen (see Figure 4-3).

1. Click on the = **sign** in the formula-bar


Figure 4-3. Example of the Textual manual


The manuals, written in Dutch, consisted of five chapters covering the topics: starting MSEXcel, creating a datasheet, changing a datasheet, creating a graph, and closing MSEXcel. Apart from the presence of the screen captures the three manuals were identical in content as well as lay-out.

Questionnaires and tests

Participants were asked to think aloud while carrying out the experimental task. This way the observers had two cues for observing switching behavior, namely the (sometimes slight) movement of the head when users looked up from the manual to the screen and the corresponding thoughts of the user. Switching attention was recorded by jotting down the number of times a user looked up from the manual to the computer screen. These notes were recorded on an observation form which contained all text fragments of the manual. That is, each instruction (action step) in this form was connected to a check box in which the observer noted the number of times the participant switched attention for that instruction. The observers were also instructed to mark those moments in which they failed to observe switching behavior (e.g., because it could not be ascertained with full certainty that the user switched attention). The two observers did a pilot observation in which they both observed two participants to reach agreement about the use of the observation instrument. In this pilot, interrater reliability (Cohen’s Kappa) was 0.58, which is considered acceptable (Eggen & Sanders, 1993; Bechger, Hemker, & Heuvelmans, 2001).

Some of the personal characteristics of the participants were measured with a short questionnaire about *gender* (male or female), *age* (in years), *touch-typing skill* (yes or no), and *computer experience* (based of self-grading on a scale from 0 (no skill at all) to 10 (extremely well-skilled)).

Training time was measured in seconds by the observer with the aid of a stopwatch. After training, participants received a written test which consisted of 5 questions. Two questions involved trained tasks and three questions asked about untrained tasks. *Trained tasks* are tasks that are the same as practiced with the manual. An example of a trained task is: ‘What happens when you click on the -icon?’. Transfer tasks or *untrained tasks* test whether participants can apply their

recently acquired knowledge and skill to (slightly) new aspects of the software. An example of an untrained task is: ‘What information is needed before one can use the -icons?’

Procedure

The experiment was held in two adjacent private rooms with the observer seated in one of these rooms and the participant in the other. Participants were told that the goal of the study was to investigate how people act when they learn to work with a computer program using a paper manual as their only form of support. Participants were seated behind a computer and asked to think-aloud during training. The observer was seated in the adjacent room which gave an unobtrusive view into the other room through a one-way mirror. During training the observer registered the participant’s actions. The training session took a maximum of 40 minutes. After a short break, the participant was asked to complete the test. Participants were not informed about this test beforehand. After the test they were informed of the goal and design of the experiment.

Coding and scoring

Number of participants in the analyses

The observation form together with the think-aloud procedure proved to be a workable format for recording switching behavior. Even so the observers were not able to score the user’s reactions to all instructions. This was mainly due to the speed with which some participants processed some of the instructions. The manuals included 45 instructions (action steps). Therefore, there were also 45 moments in which the information in the manual intended a switch between manual and computer screen. For five participants more than 10% of observations for these expected switching moments were missing. These participants were removed from the dataset, yielding a total number of 37 participants in the statistical analyses.

Switching attention

The observation data were examined in three ways. To assess the *relative number of switches* the behavior for each instruction was scored as a switch (score 1) or a non-switch (score 0), regardless of the number of times the user switched attention for that particular instruction. The relative number of switching acts is therefore simply the user’s actual score divided by the maximum number of instructions. The theoretical maximum of the latter is 45 given the 45 instructions in each manual. When there are missing values, this maximum is adapted. For example, if the observer had failed to register 2 instructions of a participant, the actual score for this person would be divided by 43.

Participants could switch more than once for an instruction, of course. Repeated switches may occur, for example, when the user rereads an instruction or wants to be absolutely sure that the screen capture and the actual screen are the same. Such repeated switching acts are taken into account by calculating the *absolute number of*

switches. This measure is simply the mean sum of all switching acts of a participant. For this measure the maximum score can lie considerably above 45 switches.

We also calculated a third measure of switching. The measure is the *mean rank order of switches*. In this measure the three conditions are given a rank score for each of the 45 instructions on the basis of the mean number of switches, where a large mean amount of switches leads to a high rank. For example, in the three conditions the mean number of switches for the first instruction were 1.25 for the Visual-Full, 1.18 for the Visual-Part&Full, and 1.36 for the Textual condition. This led to the following rank order for the first instruction: 1 for the Visual-Part&Full manual, 2 for the Visual-Full manual, and 3 for the Textual manual.

Learning effects

The two trained test items and three untrained test items were divided into distinct test elements. For each correct answer to such an element a participant received one point. As both trained test items consisted of two elements, this led to a maximum score of 4 for the trained tasks. The three untrained task items consisted of 7, 2, and 1 element(s), leading to a maximum score of 10 for the untrained tasks.

Results

Check for randomization

For the covariates computer experience and age, Anova's were used to check the random distribution of participants over conditions. For the same reason, Chi-square tests were conducted for the covariates gender and touch-typing skill. The findings show that there were no statistically significant differences between conditions on personal characteristics except for age (see also Table 4-1). Tukey tests revealed that participants in the Visual-Full condition were younger than the participants in the Textual condition ($F(1,36) = 3.962, p = 0.028$). Although this finding shows that the participants were not randomly distributed over conditions based on their age, there is no theoretical assumption that foresees a disturbing influence of this variable. Age is therefore not treated as a covariate in the analyses³.

Training time and Learning effects

There were no differences between conditions on training time and on learning effects (see Table 4-2). Most of the participants spent less than half an hour on learning how to work with the program. They scored 87% correct on trained tasks and 48% correct on untrained tasks.

³ The results on training time, learning effects, and switching attention show a similar pattern with and without age as a covariate.

Table 4-1. The participants' age (in years), gender and ability to touch-type by condition

	Age (sd)	Gender		Ability to touch-type	
		Female	Male	Yes	No
Visual-Full	20.50 (1.24)	11	1	5	7
Visual-Part&Full	20.72 (2.20)	9	2	4	7
Textual	22.21 (1.58)	12	2	3	11

Table 4-2. Means (standard deviations) on training time, learning effects, and switching attention

	Training time in seconds	Learning effects		Switching attention	
		Trained tasks (max=4)	Untrained tasks (max=10)	Percentage of switches	Absolute number of switches
Visual-Full (n=12)	1645 (317)	3.17 (1.11)	4.58 (1.56)	97.14 (2.60)	59.17 (6.93)
Visual-Part&Full (n=11)	1592 (181)	3.55 (0.69)	5.55 (2.11)	97.06 (2.58)	59.64 (11.12)
Textual (n=14)	1584 (266)	3.71 (0.47)	4.29 (2.23)	96.89 (2.76)	57.57 (6.39)

Switching attention

The behavior of participants on switching attention was tested in three ways: in a relative, absolute, and rank order manner. Table 4-2 shows the mean scores for each condition on the percentage of switches (relative) and on the amount of switches (absolute). Anova's showed no statistically significant differences between conditions for these measures. In about 97% of the possible moments in which a switch was called for and expected, users did look up from the manual toward the screen at least once. Furthermore, the mean absolute number of switches lies between 58 and 60 times indicating that the average users switches about 1.3 times for an instruction. Here too, no differences between conditions were found. The reason that this average is higher than one switch lies in the fact that some action steps in the manual contained more than one task. The peaks for the amount of switches for the actions steps 4, 7, 19, and 26 in Figure 4-4 can be explained by this. These steps asked, for example, to 'Type in cell A4 to A12, student 2 to student 10', or to 'Change the formulas in cells G4 to G12 yourself'. This is in contrast to the majority of action steps that asked the user to carry out a single task like, for example, 'Click on cell C3', or 'Select Insert from the menu bar'. Action step 24 asked to type in a rather complex formula that was hard to memorize, which also caused a need for multiple switches.

The rank order measure indicates that perhaps the conditions are not as similar as the relative and absolute measures suggest. Friedman's rank order test revealed a statistically significant difference ($\chi^2(2) = 6.671, p = 0.036$) with mean ranks 2.16, 2.14, and 1.70 for the Visual-Full, Visual-Part&Full, and Textual manual respectively. Because the mean ranks of the two visual conditions were almost identical, the two were combined into one and a subsequent Two Groups Sign test was run. This test too showed a statistically significant difference favoring the visual manuals ($z = -3.050, p = 0.002$). The exact comparison between the visual and textual manuals is 32 cases in which the rank of the Visual manual is higher than the Textual manual, 11 cases in which the Visual manual is lower than the Textual manual, and 2 cases which rank equally.

Discussion

The main aim of this study was to find empirical support for the view that screen captures stimulate users to switch attention. Although the findings from this study are not entirely conclusive in this respect, based on the results of the rank-order analyses, there are indications that a manual with screen captures may stimulate users to switch more often between manual and computer screen than does a textual manual.

For the percentage of switches a ceiling effect was found. Regardless of manual type, the users switched attention in about 97% of the instructions. In other words, users in all conditions switched attention when there was an objective need to do so. This finding calls into question the added value of a visual design. Perhaps it simply indicates that the action-oriented approach of the (minimalist) tutorials in combination with the user's computer skill suffice to realize a desirable minimum of a 'switching mode'.

The results of this study did not lead to insight in the benefits or drawbacks of a particular visual design to support switching attention. None of the tests pointed to a difference between design variants. An explanation for this may be that the full screen captures were very readable after all. If this were the case, both visual manuals become equal in use. There is no longer an advantage for the well readable partial screen captures of the Visual-Part&Full manual over the Visual-Full manual without those partial screens. At the same time, both designs become evenly self-contained, canceling out the advantage of the Visual-Full manual that was supposed to force the user to look up to the screen over the self-contained Visual-Part&Full manual.

A second explanation may be that users of the Visual-Part&Full manual ignored the redundant partial screen captures. If these screen captures were indeed not used, this neutralizes the difference in design between the two visual variants.

A third explanation may be that users adapt to the design of the manual. Although they equally benefit from the screen captures offered, their use of these visuals could have been different. More specifically, users of the Visual-Full manual

switch regularly because they are forced to do so by the unclear screen capture in the manual in comparison with the actual screen. Users of the Visual-Part&Full manual switch equally often because the combination of partial and full screen captures makes switching easy and thus convenient for them. Although the results are the same, the realization of these results may have been different. Studies on such exact picture processing may provide further insight in the effects of screen captures for switching attention between the manual and computer screen.

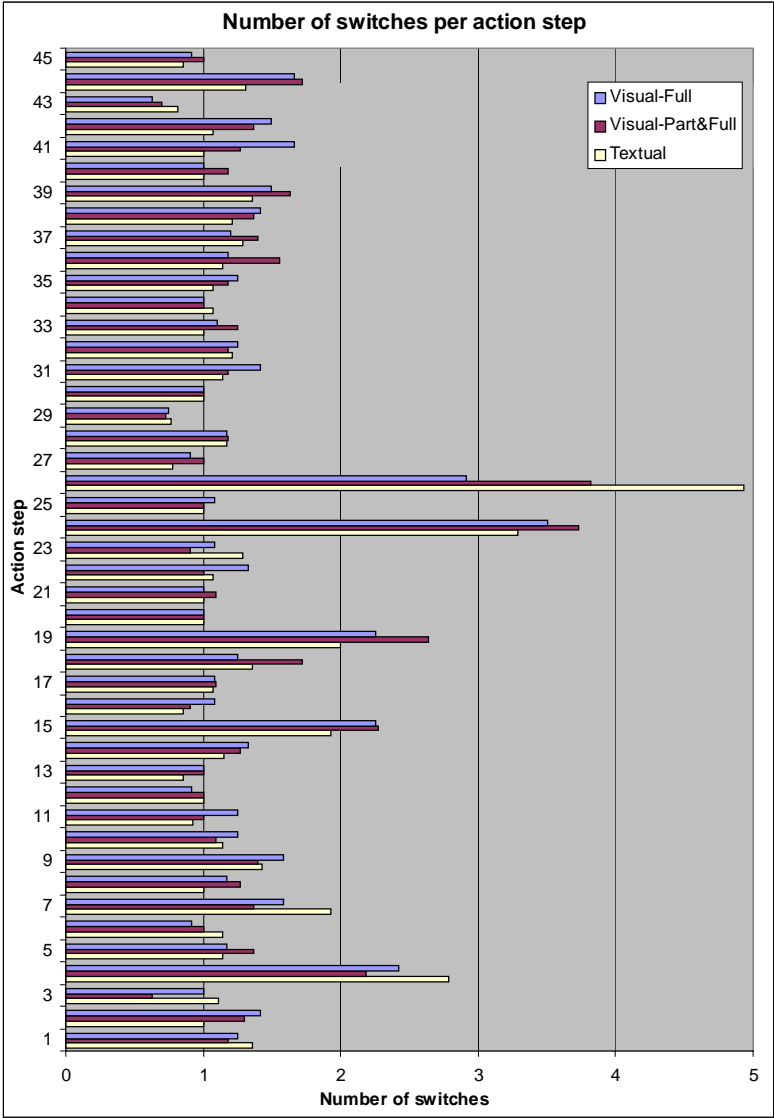


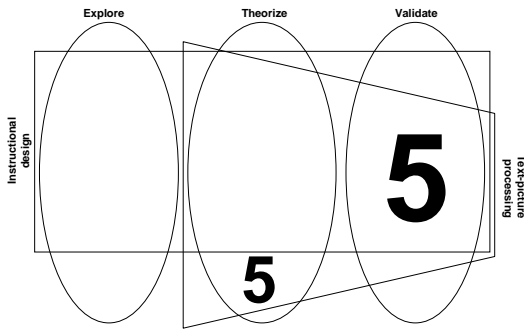
Figure 4-4. Absolute number of switches for instructions in the visual and textual manuals.

Acknowledgements

We would like to thank Chantal Blockhuis and Maaike Korte for their valuable help on preparing the manuals used in this study and for carrying out the observations. We would like thank Ton de Jong and Jules Pieters for their reviews on earlier versions of this paper.

References

- Bechger, T., Hemker, B., & Heuvelmans, T. (2001). *Het bepalen van beoordelaarsovereenstemming in een vooronderzoek voor de staatsexamens Nederlands als tweede taal en de Profieltoets [Determination of interrater reliability in an analysis for the state examens Dutch as a second language and a profile test]*. Arnhem: CitoG, Psychometrisch Onderzoekscentrum.
- Carroll, J. M. (1984). Designing minimalist training materials. *Datamation*, 30, 125-136.
- Eggen, T. J. H. M., & Sanders, P. F. (1993). *Psychometrie in de praktijk [Psychometrics in practice]*. Arnhem: Cito.
- Gardner, D. C., & Beatty, G. J. (1994). *Visuele leer methode Windows 3.1 [Windows 3.1: The visual learning guide]*. Utrecht: A.W. Bruna Uitgevers B.V.
- Gellevij, M., Van der Meij, H., De Jong, T., & Pieters, J. M. (1999). The effects of screen captures in manuals. A textual and two visual manuals compared. *IEEE Transactions on Professional Communication*, 42, 77-91.
- Sweller, J., & Chandler, P. (1994). Why some material is difficult to learn. *Cognition and Instruction*, 12, 185-233.
- Van der Meij, H. (1997). The ISTE approach to usability testing. *IEEE Transactions on Professional Communication*, 40, 209-223.
- Van der Meij, H. (2000). The role and design of screen images in software documentation. *Journal of Computer Assisted Learning*, 16, 294-306.



Validation: Developing a Mental Model; Identifying and Locating Window Elements; Verifying Screen States¹

Abstract - This study compares a multimodal instruction with text and pictures with a unimodal, text-only instruction. More specifically, forty-four students of a department of teacher education used a visual or a textual manual that helped them to learn to use a complex software application. These manuals were designed for self-study. During the 103-116 minute training sessions, cognitive load, and time and ability to recover from errors were measured. After training, we tested for learning effects on trained and untrained tasks. These variables were measured to examine three questions. The first question concerns two conflicting views on how the combination of text and pictures in software manuals is processed in memory. Dual Coding theory predicts an advantage of multimodal instruction thanks to the distinct processing of text and pictures. Cognitive Load theory, in contrast, predicts that a visual manual causes too much cognitive load because of the redundant character of the screen captures. The results for cognitive load, training time, and learning effects in the present study initially support Dual Coding theory. However, when combining the findings in this study, explanation from Cognitive Load theory remains possible. A second question is whether a multimodal approach is advantageous for complex, 'on the job' learning. The experimental task is complex because the user must deal with several information sources (manual and computer screen) and input devices (keyboard and mouse), and must handle the interaction with and between these sources and devices. The results show that even in this complex situation, the

¹ Gellevij, M.R.M., Van der Meij, H., de Jong, T., & Pieters, J.M. (2002). Multimodal versus Unimodal Instruction in a Complex Learning Context. *Journal of Experimental Education*, 70, 216-240.

multimodal instruction led to a better performance than the unimodal instruction. Finally, the study set out to validate a framework on the main roles or functions of screen captures in manuals. The results show that, based on these roles, it is possible to optimize an instructional design to strengthen learning. That is, the multimodal manual led to a stronger mental model of the computer program, improved identification of window elements and objects, and speeded up the location of window elements and objects.

Introduction

Instructional material in which the use of various media is integrated improves learning more than instruction with only one medium (Mayer, 1999; Mayer & Sims, 1994; Mayer, Moreno, Boir & Vagge, 1999; Mayer & Gallini, 1990; Robinson, Robinson, & Katayama, 1999; Kulhavy, Stock, Peterson, Pridemore & Klein, 1992). Especially Mayer and his co-authors have repeatedly shown that users benefit from a multimodal approach, whose most common form is that of a mixture of words and pictures. This study examines the value of text-picture combinations for a situation in which participants must learn to use a complex computer program through self-instruction.

The typical presentation platform of multimodal instructions in such a situation is a visual manual in which screen captures, displays of the computer screen, are combined with text (Van der Meij & Gellevij, 1998). The context is of special interest to research because the screen captures in the manual also appear on the users' monitor when they process the instructions. In this situation the findings from research following Dual Coding theory is challenged by Cognitive Load theory. While Dual Coding theory predicts that users benefit from a mixture of words and pictures due to the simultaneous processing in two distinct memory systems, Cognitive Load theory predicts that users are likely to suffer from cognitive overload due to the redundancy of the screen captures. This study aims to examine which of these two theories best fits the learning that takes place in this context.

The study is special in that it was conducted with a real world task. The possible advantage of a multimodal instruction over a unimodal one is tested in a realistic, 'on the job' learning context.

Yet another goal of the study is that it aims to validate a framework on key roles or functions of screen captures. More specifically, the study examines whether adapting the design of screen captures optimizes learning. That is, it examines the effects of fitting the (three of four) roles onto the users' tasks.

Processing text and pictures: Dual Coding

According to Dual Coding theory, the basic reason why better learning occurs when words and pictures are combined, lies in how the learner processes information in memory. Dual Coding, or dual processing, presupposes that working memory consists of two distinct systems: a verbal and non-verbal system. Using the capacity of both systems leads to *more* information being processed than when using only one of the systems. In addition, it also yields *better* results because the simultaneous processing renders the connectivity of the two systems. This referential connectivity in turn contributes to the construction of a strong mental model (Paivio, 1990; Mayer, 1999). That is, it leads to "...a model that evolves in the mind of a user as he or she learns and interacts with a computer system. ... It represents the structure and internal relationships of a system ... and is the source of the user's expectations about

the effects of actions; it can guide navigation or planning of actions and contribute to interpretation of feedback” (Jih & Reeves, 1992, p. 45)².

Processing text and pictures: Cognitive Load

Working memory is limited. When people face a task that is already quite difficult, ‘additional’ instructions may be more of a burden than an aid. This, in short, is a key tenet of Cognitive Load theory. The theory may prove to be valuable for the present study because the experimental task is complex. One of the complexing factors is that learning to use a computer program requires the use of more information sources and the handling of more devices than just the instructional material. Users must not only process the manual, they must also attend to the keyboard, mouse, and computer screen. Cognitive Load theory indicates that this situation poses two potential risks: redundancy and split attention effects (Chandler & Sweller, 1991; Sweller, 1994; Sweller & Chandler, 1994).

Redundancy occurs when the manual presents screen captures that the user can also see on the computer monitor. The *redundancy hypothesis* predicts that offering the same information twice, as in the case of a depicted computer screen, requires the user to process this information twice. The double processing takes up unnecessary memory space, space that could have been used to process other information. According to Sweller and Chandler, redundant information is therefore likely to increase training time and decreases learning effects.

In a series of experiments to test the redundancy hypothesis, Chandler and Sweller (1991) found that when a picture can be understood all by itself, adding explanatory text was redundant and decreased performance. A similar result was found in another experiment of Sweller and Chandler (1994). In that study they compared the use of a text-plus-picture (screen capture) manual when learning how to use a computer program with and without the use of the computer. Here they found that *not* using the computer lead to the best learning because the information on the computer screen was redundant. The pictures in the text-plus-picture manual already offered the necessary information.

Split attention effects occur when people must attend to different sources of information simultaneously. The *split attention hypothesis* predicts that in such a situation learning is hampered when these sources must be integrated. According to Sweller and Chandler, split attention effects obstruct learning when a text fragment is needed to understand a picture and vice versa because the user then has to process two distinct information sources at the same time. Split attention effects occur only when text and picture in the manual are mutually dependent. If text and picture can be understood independently, users are likely to opt for one of the two presentation modes.

² Jih and Reeves derived this definition from Gentner & Stevens, 1983; Norman, 1983; and Van der Veer, 1989.

For visual manuals the separation between the information sources, text and picture is small. Sweller and Chandler (1994) also examined the presence of split attention effects in a situation with a stronger physical separation. That is, they compared a text-plus-picture manual *without* using a computer with a textual manual *with* the use of a computer. They found a negative effect of split attention in the latter situation because the user had to switch attention between computer and manual. The manual with text and pictures yielded good results. This is probably because in this manual text and picture were integrated preventing redundancy effects. Text nor picture could be understood without the other.

Processing text and pictures: Cognitive Load or Dual Coding?

According to Cognitive Load theory, the use of a multimodal manual, even when text and pictures are integrated, is *not* a preferable instructional format compared to a unimodal instruction when using the manual and the computer at the same time. The screen captures are redundant because the same information is already presented on the computer screen, increasing the load on working memory, and are thus likely to increase training time and decrease learning. In contrast, Mayer's experiments show that a multimodal approach of text-picture combinations exceeds text-only instruction. Thanks to the distinctly different nature of text and pictures and their separate processing of content similar information, the information is better integrated, resulting in a stronger mental model and, more generally, strengthened learning.

The above contrast only stands when processing the content to be learned requires the use of memory space to a large extent. Sweller (1994) divides cognitive load into two types: intrinsic cognitive load, caused by the content of the instruction, and extraneous cognitive load, caused by the design of the instruction. Intrinsic cognitive load is caused by element interactivity. This element interactivity is a critical element in Cognitive Load theory. When elements can be learned in isolation, there is low element interactivity. As a result, intrinsic cognitive load will be low, leaving enough memory space for extraneous cognitive load. In such a case, the quality of the designed instruction is not a critical factor because there is enough memory space to compensate for any possible flaws. The quality of design only matters when element interactivity is high and the available space for extraneous cognitive load is (very) limited. Reducing extraneous cognitive load is therefore beneficial only when intrinsic cognitive load is high (Sweller & Chandler, 1994).

In this study the complexity of the program was high. The content had high element interactivity, causing high intrinsic cognitive load. For example, (1) an action step was meaningless when considered in isolation. Series of (mutually dependent) action steps served as meaningful entities, (2) sections of explanatory texts became meaningful in combination with the accompanying action steps, and vice versa, and (3) multiple series of action steps together formed meaningful procedures.

Because of the high intrinsic cognitive load of the experimental task and the redundant character of using the computer screen in combination with a visual

manual, learning may be hindered. In the next section we give suggestions for the use of screen captures in manuals to maximally reduce extraneous cognitive load caused by the design of the instruction.

Instructional Functions of Screen Captures

Research on the use of screen captures in software documentation has yielded varied results. In some studies, a multimodal manual has been found to yield better results than did a unimodal (Sweller & Chandler, 1994; Van der Meij, 1996). Other studies found no difference or a better result for the textual manual compared to one type of visual manual (Nowaczyk & James, 1993; Gellevij, Van der Meij, de Jong, & Pieters, 1999). The difference can perhaps be explained by examining these studies from the stance of Cognitive Load theory and Dual Coding theory. In addition, we suggest that it is advantageous to use a dedicated theory on screen captures to supplement the more general views on memory processing. Below we examine the variations between studies and suggest how screen captures can best be presented in manuals.

To date, only a few experiments on screen captures have been conducted. Besides our own research (Van der Meij, 1996; Van der Meij, 2000; Gellevij et. al., 1999), we found only two other studies (Nowaczyk & James, 1993; Sweller & Chandler, 1994). These studies differ rather strongly on a number of critical factors. One, the *designs of the screen captures* vary tremendously. For example, the depicted screens represented the full range of possibilities for presentation in the manual. Namely, pictures of the full screen, of partial screens, and of icons. Two, for *positioning text and picture*, the text was placed either on the left-hand side, the right-hand side, or to either side of the picture. Three, the *type of manual* varied; in one study the instructions were given in the form of a job aid, whereas in others rather extensive tutorials were used. As a result, the *magnitude of the task* ranged from short and simple to long and complex. Four, there were additional variations on the *experience of the user group* (novice vs. experienced users), and the possibilities of using the application during training.

To regulate research and practice, Van der Meij and Gellevij (1998) have proposed four functions of screen captures. In line with Mayer's SOI-model of constructivist learning from words and pictures (1999), these functions focus on maximizing the user support for selecting, organizing, and integrating information. In the framework of Van der Meij and Gellevij (1998), the main function for screen captures is supporting the development of a mental model. The other functions in the framework are subordinate, contributing in different ways to mental model development. These other functions are: switching attention, verifying screen states, and identifying and locating window elements and objects.

Developing a mental model of the program. Each computer program (or series of computer programs) has its own look-and-feel that influences the development of a mental model. This mental model helps users explain how the program works and allows them to predict what happens when they carry out a certain action. Screen

captures can help users develop such a mental model by acquainting them to the interface, by showing how windows are arranged, and by showing how the interface changes during task execution.

Switching attention. When learning to operate a computer program with the use of a manual, users often fall into the trap known as the nose-in-the-book syndrome. Users should be attended to the moments in which they should look up from the manual and examine the computer screen. The manual should stimulate users to look back and forth from the manual to the screen regularly. Screen captures can prompt the user to do so. This function is not further detailed in this paper because it is not tested empirically.

Verifying screen states. Especially novice users may fear that their actions will damage the computer or the application and therefore are reluctant to work on the computer unimpeded. The manual can alleviate these fears by presenting information that allows users to verify that they are on the right track. For more experienced users verification information remains important. It provides positive feedback and reinforces motivation when the information shows they are on target. In addition, it can help them detect errors early on and thus facilitate error management. Screen captures are optimally suited for this verification process. Comparing a picture of the screen with the actual screen is easier than comparing the content of written text with a screen.

Identifying and locating window elements and objects. Interfaces nowadays show so much information that users are faced with two problems: they must learn the meaning of specific icons and buttons and they must acquaint themselves with their position on the screen. Screen captures can help solve both problems. They can be used to explain screen elements, and they can help users focus on the relevant part of the screen.

The current study compares a visual manual with a textual manual. The visual manual contains screen captures that are designed to optimally support three of the four key functions of such pictures³. The study tests the following predictions.

No differences are expected for *cognitive load* because the visual design of the instructional material is seen as functionally redundant instead of dysfunctional. Thus, although there is overlap between the screen capture and the actual screen this display is expected *not* to impose extraneous cognitive load.

Users of the visual manual are also expected to *develop a better mental model* than users of the textual manual for two reasons. One, text and picture can more easily be integrated because they appear on the same medium (i.e. manual). Two, in contrast to monitor displays, the visual manual continuously displays a series of images after each other. These images can therefore be studied in conjunction which may help the user in building a stronger visual image of the program.

Users of the visual manual are expected to *verify screen states* faster and be more capable in recovering from errors than users of the textual manual thanks to the

³ The function *switching attention* is not tested in this study.

ease with which the displayed screens in the manual can be compared with the computer screen.

The same is true for *identifying and locating* window elements. Users of the visual manual are expected to perform better in locating and identifying screen elements and to complete training faster thanks to the visual support provided by the screen captures.

Method

Participants

Forty-four students (29 males and 15 females) from a department of teacher education participated in the study. This user group was selected because of their relatively high level of knowledge on physics (on which the case used was based), an intermediate to high level of general computer knowledge, and a low level of domain-specific knowledge of the computer program to be learned. The presence of a minimum level of domain-specific prior knowledge is seen as a necessary condition to benefit from text-picture combinations in instruction (Mayer & Sims, 1994; Mayer & Gallini, 1990). The study was presented as an introductory course that the participants could partake in as an extra-curricular activity. Participants were randomly assigned to experimental conditions.

Materials

Computers

The sessions were held in three different computer rooms with 16 to 20 IBM compatible computers. During the experiment, the participants' actions with the computer program were automatically logged.

SimQuest and Motion application

The computer program SimQuest (de Jong, et. al., 1998) consists of two parts: a learner part (application) and authoring part. SimQuest applications offer learners discovery learning environments in which they can explore a specific domain. Their discovery behavior is evoked through simulations. To support the discovery process, learners receive assignments and explanations in forms such as video, (html-)texts, pictures, and sound.

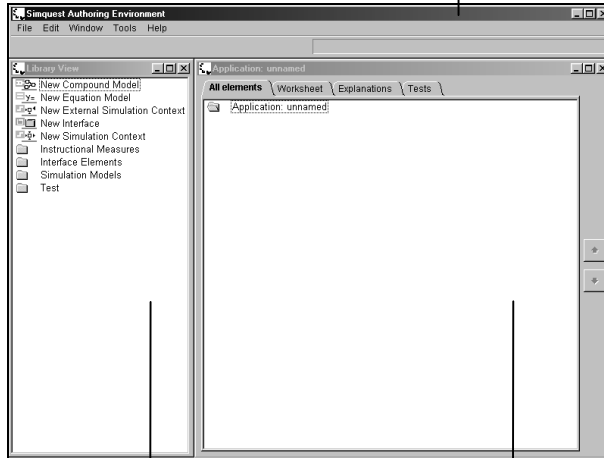
Teachers or instructional designers use the SimQuest authoring environment to modify an existing learner environment or create a new one. SimQuest is based on an object-oriented approach, which means that a collection of ready-made elements can be used to create an application or program. In the reported study, participants learned to work with the SimQuest authoring environment.

The participants were expected to be interested in learning how to use the SimQuest software. As future teachers they were informed about the use of a new form of instructional material. As future designers of their own lessons, they were informed about the possibilities of modifying and creating learning environments through SimQuest's authoring environment.

Visual manual

SIMQUEST menu and status

With this window you can operate the general functions of SIMQUEST, like saving an application or exiting SIMQUEST. This window also shows the status of operations SIMQUEST is performing, for example the loading of an application.

***Library***

The library window contains standard elements you can use to create a new application.

Application

The application window shows the elements that you have put into your application. The application window also shows the structure of your application.

Textual manual

SIMQUEST menu and status (window on top)

With this window you can operate the general functions of SIMQUEST, like saving an application or exiting SIMQUEST. This window also shows the status of operations SIMQUEST is performing, for example the loading of an application.

Library (window bottom left)

The library window contains standard elements you can use to create a new application.

Application (window bottom right)

The application window shows the elements that you have put into your application. The application window also shows the structure of your application.

Figure 5-1. Screen capture design in **explanatory sections** of the visual manual, for the functions *developing a mental model of the program and identifying and locating window elements and objects*, and the design of the textual counterpart (translated from Dutch)

The SimQuest application used to exemplify the use, modification, and creation of a discovery learning environment in the manuals, dealt with the topic of motion in physics. In the motion application, students can explore the relationships between initial velocity, velocity at a certain point in time, and acceleration. Participants are shown various simulations with moving motorcycles, trains, cars, scooters, etcetera. Assignments allow the student to check the correctness of discovered relations between variables. Explanations such as videos and textual information introduce and discuss the variables in the simulations and assignments. The manual focused on the basic tasks in handling SimQuest and covered the modification and creation of interfaces and assignments.

Manuals

The design of the screen captures has an important influence on its potential effects because function and form interact. The following designs were used in the study.

The designs for *developing a mental model* were based on two major features of illustrations that support learners in building a runnable mental model as described by Mayer and Gallini (1990). These two features are *system topology* and *component behavior*.

System topology means that the important elements of the screen are shown and explained within their immediate context. In this case the ‘important elements’ are windows, parts of windows, icons and buttons. The ‘immediate context’ is represented by a screen capture of the complete or full computer screen. System topology is mainly applied in explanatory sections of the manual (see Figure 5-1).


Component behavior means showing “each major state that each component can be in and the relation between a state change in one component and state changes in other components” (Mayer & Gallini, 1990, p. 715). Applied to the design of screen captures in the procedural sections (action steps) of the manual, this means showing how the computer screen changes as a result of carrying out actions with the various components or elements of the computer program (see Figure 5-2).

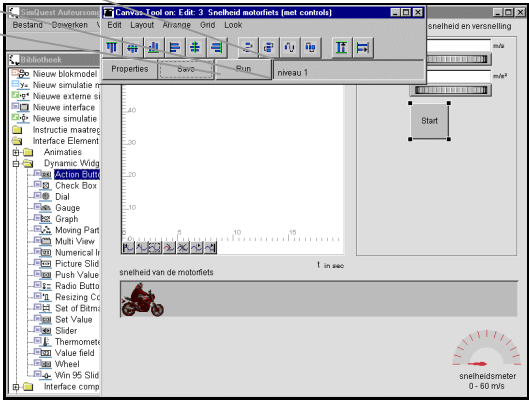
For the function *verifying screen states*, the design arguments were: (1) providing a size that is legible, to make sure that the *specific content* can easily be verified, (2) only showing the relevant part of the screen to help focus attention, and, (3) if necessary, using a cueing technique to point to the essential verification part (see Figure 5-3). Note that the layout is basically the same as for mental model development. The flow of action steps and accompanying screen captures stays intact to show component behavior and to keep the design consistent throughout the manual. The essential difference is that *not* the full screen is shown, but only the relevant window to improve legibility.


The design for the function *identifying and locating window elements and objects* was mainly based on considerations of system topology and component behavior. In addition, cueing was used to point to the essential screen parts, and callouts were used to explain those parts. For locating, providing context is important to find the relevant screen part quickly. For identification, context is of less importance. Locating and identifying window elements through screen captures can take place in both procedures and explanations. Figure 5-2 shows an example of

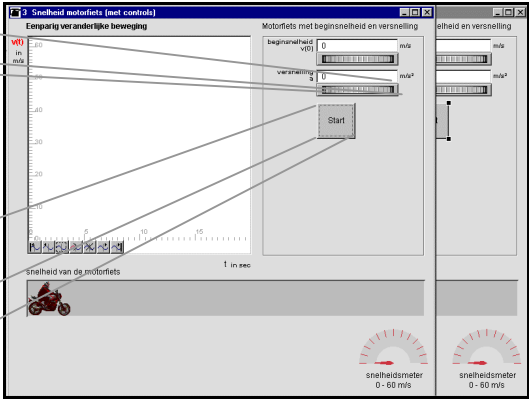
the design for a procedure. Figure 5-1 shows the design in an explanatory section. Note that these designs are the same as for mental model development. These screen captures thus serve more than a single function.


Visual manual

1 Click on 



2 Give values by 'turning' the 



3 Click on 

Textual manual

- 1 Click on **Run** in the window *Canvas Tool*
- 2 Give values by 'turning' the **Wheel**
- 3 Click on **Start**

Figure 5-2. Screen capture design in **procedural sections** of the visual manual, for the function *developing a mental model of the program and identifying and locating window elements and objects*, and the design of the textual counterpart (translated from Dutch)

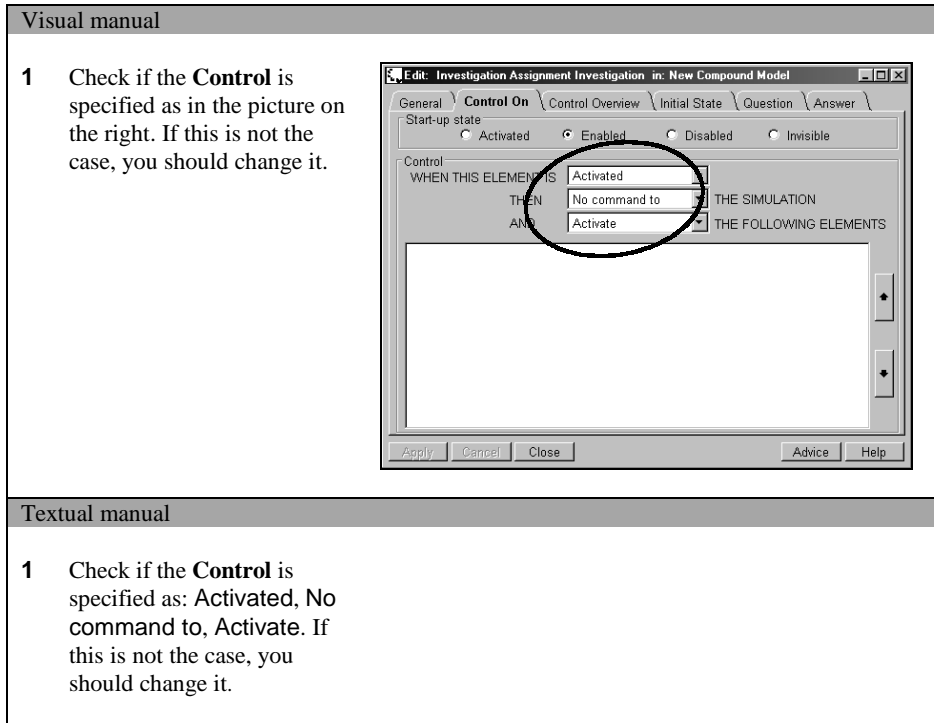


Figure 5-3. Screen capture design for the function *verifying screen states*, and the design of the textual counterpart (translated from Dutch)

The two manuals (textual and visual), which were written in Dutch, consisted of three chapters:

- Chapter 1: Introduction, in which basic operations like starting SimQuest, loading an application, handling the different windows, etcetera are treated, and in which the basic structure and components of the system are explained;
- Chapter 2: Modifying and creating interfaces, in which an interface to present and manipulate the values of the model variables is created;
- Chapter 3: Modifying and creating assignments, in which variants to promote and practice with discovery behavior are explained and created.

Chapters 1 and 2 of the visual manual were designed to support the development of a mental model and to further assist the user in locating and identifying window elements and objects. The difference between the textual and visual manual resided in the absence of screen captures in the textual manual. Chapter 3 of the visual manual was designed to support the process of screen verification. Again, the difference between the visual manual and its textual variant resided in the absence of screen captures in the textual manual. The lay-out and typography of the visual and textual manual were further as similar as possible to avoid differences in reading and to make sure that any differences could be attributed to the screen captures.

The presence of screen captures obviously led to manuals of different sizes. Table 5-1. shows the number of pages for each chapter of the manuals and the number of screen captures used for each of the distinguished screen capture functions. Table 5-2 shows the number of screen capture types used.

Table 5-1. Number of manual pages and number of screen captures in the textual and visual manual

	Manual	
	Textual	Visual
Chapter 1+2+3 Number of pages	66	96
Chapter 1 Number of pages	13	22
Number of screen captures		
Mental model, Identifying & Locating		
Procedures (Figure 5-2)	0	22
Explanations (Figure 5-1)	0	4
Verifying screen states		
(In procedures only (Figure 5-3))	0	0
Chapter 2 Number of pages	22	39
Number of screen captures		
Mental model, Identifying & Locating		
Procedures (Figure 5-2)	0	49
Explanations (Figure 5-1)	2 ¹	3
Verifying screen states		
(In procedures only (Figure 5-3))	3 ²	3 ²
Chapter 3 Number of pages	31	35
Number of screen captures		
Mental model, Identifying & Locating		
Procedures (Figure 5-2)	0	0
Explanations (Figure 5-1)	0	1 ³
Verifying screen states		
(In procedures only (Figure 5-3))	0	30

Note. Screen capture designs in procedures for mental model development & Identifying and Locating consist of a full screen captures plus one or more captures of icons, buttons or objects (see Figure 5-2).

¹ The design of the interface to be created by the students could very hard be described in words. It was considered necessary to show this with screen captures, even in the textual manual.

² Creating a new interface was a complex task. To keep users on the right track, it was considered necessary to provide feedback in the form of verification information, although this was *not* the intention in this chapter of the manual.

³ The unintended presence of this screen capture was caused by a copying error. It is considered to have had no influence on the results.

Table 5-2. Number of screen captures types in the textual and visual manual

	Manual	
	Textual	Visual
Chapter 1		
Procedures (32 action steps)		
Full	0	22
Window	0	0
Icon/button/object	0	26
Explanations		
Full	0	3
Window	0	3
Icon/button/object	0	0
Chapter 2		
Procedures (86 action steps)		
Full	0	49
Window	3 ¹	3
Icon/button/object	0	78
Explanations		
Full	0	0
Window	2 ¹	3
Icon/button/object	0	0
Chapter 3		
Procedures (124 action steps)		
Full	0	3
Window	0	27
Icon/button/object	0	0
Explanations		
Full	0	0
Window	0	1
Icon/button/object	0	0

¹ The design of the interface to be created by the students could very hard be described in words. It was considered necessary to show this with screen captures, even in the textual manual.

Questionnaires and tests

Participants received a paper questionnaire with general questions about their name, age, topic of study and grade level. In addition, there were 11 questions to determine the participant's *computer experience*. For this purpose, items were taken from the Computer Self Efficacy Scale questionnaire (Murphy et. al., 1989) and translated into Dutch. The questionnaire uses a 5-point Likert scale. Examples of items used are: 'I feel confident understanding terms/words relating to computer software', 'I feel confident learning a variety of computer programs', and 'I feel confident explaining why a program will or will not work on a given computer'.

While working with the manual, participants encountered some pages that asked them to fill in the current *time* (which was displayed at the computer screen) and to rank their current *cognitive load*. Training time is a general measure for the ease with which the information in a manual can be processed (assuming that learning will not be negatively influenced by shortening of training time). This ease is mainly caused by the convenience of coordinating between the manual and the computer screen. More specifically, the support given by the manual for locating a screen object or screen element directly impacts on the time needed to complete the instruction.

To measure cognitive load, participants were asked ‘How difficult would you rate the task at this moment?’. Their answers could range from ‘extremely difficult’ to ‘extremely easy’ on a nine-point Likert scale. Paas, Van Merriënboer, and Adam (1994) have shown this to be a reliable and sensitive way to measure cognitive load. They compared this subjective measure with an objective one, namely heart rate variability. With heart rate variability, they were able to show only differences between mentally active and inactive periods. For studies such as the present one, measurement of cognitive load needs to be sensitive enough to register relatively small changes in mental effort. In two studies, Paas, Van Merriënboer, and Adam proved that the 9-point rating scale technique could detect “differences in type of practice problem and variability in the training conditions, the effects of the training condition on transfer performance, and two levels of task complexity” (Paas, Van Merriënboer, & Adam, 1994, p. 428). As an alternative, a secondary task measure could have been used. A drawback of this measure is interference with task execution. In the present study, such interference is relatively small, as the measures of time and cognitive load are taken directly after a rounded-off action steps or (sub)tasks. In Chapter 1, time and cognitive load were measured four times, Chapter 2 had seven measurements, and Chapter 3 contained eight.

After each chapter, the participants received a paper test. This test intended to measure their *mental model* and to measure whether they were capable to *identify* and *locate window elements and objects*. The distinction between the functions mental model development and identifying and locating is functional, but also artificial. It helps to design and formulate instruction because information can be presented exactly in accordance with its primary goal or function. It is artificial for measuring effects of such functions. Developing a mental model of the program always involves some identifying and locating and vice versa. To test mental model development users were asked to name or draw general windows and screens, to predict successive screens, and to point to errors in screen displays. To test identification and localization capabilities, users were asked to name specific elements, describe the meaning or function of icons, button, and screen parts, and were asked to determine the correct place of elements.

The items in these tests differed in the mode the answer was expected. Some items asked participants to describe the answer in text, while other items asked them to draw the answer in a picture. This mode of representation was balanced for the whole test to eliminate encoding differences (textually or visually) as a disturbing factor. The test consisted further of items that measured trained tasks (exercises that

were the same as practiced during training), and items that measured untrained tasks (new tasks that were not practiced or discussed in the manual).

Effects of screen captures on *screen verification* were measured as follows: (1) the time used for verification (from the log-registration), (2) the number of errors made, and (3) the rate of error recovery. To afford such error verification the manual contained three intentional errors. The information that the user must verify in the screen capture, or text, deliberately differs from the correct situation at that moment. An example of an intentional error is given in Figure 5-3. Instead of ‘Activated, No command to, Activate (as shown in the manual) the screen displays ‘Activated, No command to, Close’.

Procedure

The experiment was held in the school for teacher education. Each participant attended two sessions that lasted about two hours each. Sessions were scheduled directly before or after the regular courses and were held in two successive weeks.

Participants were seated in front of a computer where they found a short introduction sheet, the paper questionnaire, and the first chapter of the manual. The introduction sheet told participants that they should learn to work individually and independently with SimQuest using the manual as their only source of information. They were allowed to ask questions only when they encountered problems that they could not solve with the manual. The introduction also stressed the importance of filling in the time/cognitive load sheets in the manuals.

When a participant finished Chapter 1, the test for Chapter 1 was given, along with an instruction to ‘try to perform as best as they possibly can, even when some questions concern issues that they have not recently practiced with the manual’. Participants had no access to the manual or the computer during the test phase.

After finishing the first test, Chapter 2 of the manual was given, again followed by a test. After finishing this test, participants were asked to return for the second session. The second session resembled the first in set-up. Participants received Chapter 3 of the manual and made the corresponding test after finishing this chapter. After completing the experiment, the participants received the original SimQuest software and example applications, together with a (visual) SimQuest manual.

Coding and scoring

Number of participants in the analyses

All forty-four participants filled in the questionnaire on computer experience and the other general topics. Also, all participants worked on the first chapter of the manual. Four participants (two in both conditions) missed Chapter 2 of the manual due to obligatory classes. Four other participants (two in both conditions) missed Chapter 3 of the manual, some because of illness and others again due to curricular obligations. Consequently, in each condition for Chapter 1 the data of 22 participants was used in the analyses. There were data of 20 participants in each condition for Chapters 2 and 3. There were data of 18 participants in each condition when adding the results of all three chapters.

Computer experience, age, gender, topic of study, and grade level

Means of the scores on the computer experience questionnaire were calculated to determine participants' level of computer experience. Data on age, gender, topic of study and grade level could be used in the analyses untransformed.

Cognitive load

For Chapter 1, all four cognitive load measurements were used to test for differences between the two conditions. For Chapters 2 and 3, the first (which was, unlike in Chapter 1, taken before starting the task) and last measurements were removed. The last cognitive load measurement was removed because the part of the task treated in the manual before this cognitive load measure was taken unfortunately turned out to require interference of the experimenter. More specifically, the last step of the task was saving the work. As the saving operation appeared to be technically problematic, the experimenter needed to intervene, leading to an unreliable last measurement of individual participant's cognitive load. Consequently, for Chapters 2 and 3, the number of cognitive load measurements used in the statistical analyses was 5, respectively 6.

Training time on manual chapters

The time-writing sheets from the manual-chapters provided information on the training time that was used on the various chapters. For Chapter 1 the end time was subtracted from the start time. For the Chapters 2 and 3 the penultimate time measurement was subtracted from the start time.

One user of the visual manual forgot to write down the last necessary training time measurement of Chapter 2. Consequently, data of 19 participants was used for the visual condition when analyzing training time on Chapter 2.

Verification

The time used by participants to verify whether the screen showed the correct information as what was shown or described in the manual, was extracted from the log-files of Chapter 3. The time of the first identifiable step *before* the actual verifying was subtracted from the time of the first identifiable step *after* verifying. There were 17 moments where verifying behavior was triggered. These 17 time-parts were added as one total time on verification.

When determining time on verification, problems were encountered that made it sometimes difficult to be absolutely sure of the behavior of a participant. Only data that gave a clear view on the users' behavior was used in the analyses for time, resulting in data of 10 participants for the textual condition and data of 16 participants for the visual condition.

The verification information presented in the manual did, in three cases, not address the same information as on the screen. These intentional errors were put in the manual to see whether verification information was indeed used, and to see whether users in the visual condition were better in rectifying these errors. Rectifying the errors involved changing settings and adding elements. Changing to correct settings and choosing the right elements were both scored as one point.

Learning effects

Each correct element of an answer at the test after each chapter furnished one point. Examples of such elements are: a description of a single step, a drawing of a screen part, pointing to a specific error, explaining an object, or naming a location. Answers varied in number of answer-elements. Therefore, the number of maximal points for each question differed. Table 5-3 shows the number of questions used in the tests and the maximum number of points.

Table 5-3. Number of questions and maximum number of points in the tests

	Number of questions		Maximum number of points	
	Trained	Untrained	Trained	Untrained
Chapter 1				
Mental model	4	1	26	4
Locate	6	3	6	3
Identify	6	3	6	3
Chapter 2				
Mental model	5	3	22	14
Locate	7	4	8	4
Identify	10	4	14	4
Chapter 3				
Mental model	4	2	16	5
Locate	2	3	2	3
Identify	9	3	9	3

Results**Check for randomization**

To check the random distribution of participants over the two conditions, Anova's were used comparing the two conditions on the covariates computer experience and age. Chi-square tests were used on the covariates gender, major (physics, mathematics, biology), and grade level. Results of these analyses showed a random distribution on all variables.

Furthermore, it was checked if the distribution of participants' computer experience was comparable over conditions. Results show that both groups have variance on the computer experience scale, which means that users with different levels of computer experience participated in the study. The results also show that the distribution in both groups is comparable (Mann Whitney $U = 227$, $Z = -0.353$, $p = 0.724$). These findings indicate that distribution was indeed random over the two conditions.

Cognitive load

Cognitive Load theory predicts that the visual manual would impose a higher cognitive load on the user in view of the redundant nature of the screen captures. Earlier we advanced arguments that led us to predict no difference in cognitive load for the two manuals.

A one-way Anova showed no differences between conditions on the cognitive load of users while working with the manual (see Table 5-4). Multivariate Repeated Measures analyses pointed to an interaction between cognitive load and condition in the first chapter ($F(3,32) = 6.081, p = 0.002$). Figure 5-4 shows that, contrary to expectations from Cognitive Load theory, the users of the visual manual have a lower cognitive load. Interactions were absent for the Chapters 2 and 3.

Table 5-4. Means (standard deviations) for cognitive load

	Mean (standard deviation)
Chapter 1+2+3	
Textual	3.30 (1.02)
Visual	3.65 (0.57)
Chapter 1	
Textual	3.33 (1.22)
Visual	3.09 (1.10)
Chapter 2	
Textual	3.50 (1.04)
Visual	3.79 (0.70)
Chapter 3	
Textual	3.37 (1.04)
Visual	3.73 (0.92)

Note. The theoretical range for cognitive load was 0 (very, very, very easy) to 9 (very, very, very difficult). The actual range used was 0-6.

Training time

Comparing training time of the textual and visual manual resulted in a statistically significant difference in favor of the visual manual (see Table 5-5). Users of the visual manual completed training 11% faster than did users of the textual manual. According to Peeck (1993), the effects that one should expect from presenting screen captures in a manual are moderate to high. The effect sizes for training time range between 0.74 – 0.88. As an effect size of 0.25 is generally seen as a small effect, 0.50 as a medium sized effect, and 0.75 as a large effect (Cohen, 1962), the effect sizes for training time can be considered fairly high.

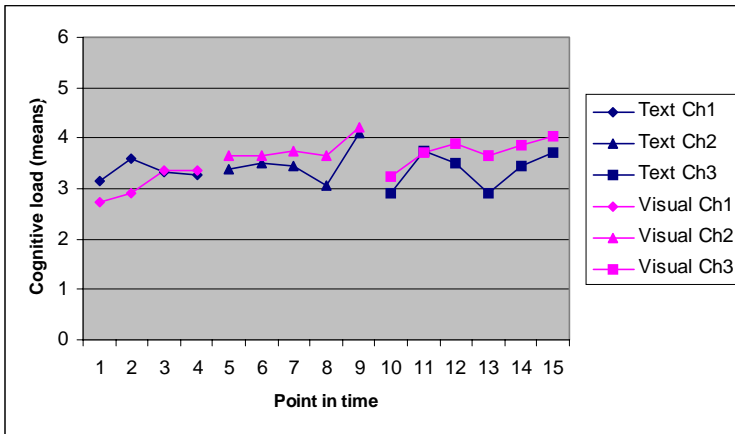


Figure 5-4. Cognitive load in the course of time

Table 5-5. Means (standard deviations) for training time in minutes

	Mean (standard deviation)
Chapter 1+2+3	
Textual	115.61 (17.26)
Visual	102.83 (15.61) ¹
Chapter 1	
Textual	23.23 (3.78)
Visual	19.91 (3.80) ²
Chapter 2	
Textual	41.15 (6.56)
Visual (n = 19)	35.58 (7.27) ³
Chapter 3	
Textual	52.45 (11.69)
Visual	48.10 (7.09)

Note. Statistically significant results are printed bold

F = Anova;

EMS = Mean Square Error value;

ES = Effect Size ((mean₂ – mean₁) / standard deviation₁);

Eta² = Proportion of Explained Variance

¹ F(1,34) = 5.425, p = 0.026; EMS = 270.85; ES = 0.74; Eta² = 0.14

² F(1,42) = 8.426, p = 0.006; EMS = 14.73; ES = 0.88; Eta² = 0.17

³ F(1,37) = 6.324, p = 0.016; EMS = 47.82; ES = 0.85; Eta² = 0.15

Verification

There were two measures to determine whether the screen captures meant to support the process of verifying *screen states* had their intended effects. One measure was the time used for verification steps. The other measure related to the detection and correction of the aforementioned ‘intentional’ errors.

Table 5-6 shows the time used on verification. Total time can be divided in time used on ‘normal’ verification steps and in verification of screen captures that were deliberately wrong and thus required correcting the error. One-way Anova’s show no differences between manuals.

Table 5-7 shows the scores for executing the corrections after comparing the verification information in the manual with the information on the screen. One-way Anova’s reveal no differences on these scores, clearly because of ceiling effects. Eighty-eight to 97% of the scores were correct, leaving little room for statistically significant differences.

Table 5-6. Means (standard deviations) for time used on verification in seconds

	Mean (standard deviation)
Verification time (total)	
Textual (n = 10)	1275 (323)
Visual (n = 16)	1213 (226)
Verification time on correct information	
Textual	97 (66)
Visual	74 (23)
Verification time on incorrect information	
Textual	1178 (291)
Visual	1140 (217)

Table 5-7. Means (standard deviations) on correcting errors (range 0-3)

	Mean (standard deviation)
Correcting elements	
Textual	2.85 (0.37)
Visual	2.90 (0.31)
Correcting settings	
Textual	2.65 (0.59)
Visual	2.85 (0.49)

Learning effects

This section reports the influence of screen captures on the *outcomes* of the instruction. These learning effects were measured with tests that determined the users' mental model of the program, their capability to identify, and to locate window elements and objects.

A one-way Anova on the total test scores shows a statistically significant overall effect on trained tasks (see Table 5-8). Users of the visual manual scored 14% higher than users of the textual manual. One-way Anova's for the specific functions show that users of the visual manual scored significantly higher on questions that measured the strength of their mental model and were significantly better in identifying window elements and objects. Users of the visual manual have gained superior knowledge of the structure of the program, have become more capable of explaining how the program works, are better in predicting what will happen when carrying out certain actions, and have gained more conceptual knowledge of the program. The effect sizes for learning effects range between 0.63 – 0.89. Similar to the results for training time, pursuant to Peeck (1993), these effects show a large potential for the visual design.

A one-way Anova showed no statistically significant difference on correctly *locating* window elements and objects. In contrast, there was a statistically significant effect of training time on locating favoring the visual manual (see section on training time).

Discussion

This study aimed to address three main questions: (1) which of two contradictory theories on information processing, Dual Coding and Cognitive Load, explains best how information taken from multimodal instruction with text-picture combinations is being processed, (2) does the superiority of the multimodal approach also hold in case of a realistic of 'on the job' self-instruction, and (3) does a framework of functional roles for screen captures in software manuals contribute to higher and more efficient learning? These issues will now be discussed in reverse order.

The framework from which the visual manual was developed has contributed considerably to its success. The manual was developed from an instructional, functional perspective in which a dedicated theory on how to optimize the presence of screen captures in a software manual was applied. Tests on the effects of the functions of screen captures yielded convincing evidence favoring the visual manual.

Table 5-8. Means (standard deviations) (percentage of maximum score) for learning effect test scores for trained tasks

	Total test	Mental model	Locate	Identify
Chapter 1+2+3				
Textual	54.06 (11.22) (50)	31.17 (7.90) (49)	10.00 (2.47) (63)	12.89 (2.89) (44)
Visual	61.83 (8.93) (57) ¹	36.17 (4.55) (57) ²	10.33 (1.97) (65)	15.33 (4.19) (53) ³
Chapter 1				
Textual	20.64 (5.03) (54)	12.09 (4.28) (47)	5.23 (0.87) (87)	3.32 (1.04) (55)
Visual	23.00 (4.30) (61)	13.82 (3.61) (53)	5.32 (0.65) (89)	3.86 (0.89) (64)
Chapter 2				
Textual	20.20 (5.80) (47)	11.75 (3.48) (53)	3.85 (1.76) (48)	4.60 (1.70) (33)
Visual	21.60 (5.97) (50)	13.05 (2.31) (59)	3.55 (1.96) (44)	5.00 (2.73) (36)
Chapter 3				
Textual	13.05 (3.27) (48)	7.05 (2.52) (44)	1.05 (0.51) (53)	4.95 (1.47) (55)
Visual	15.95 (3.47) (59) ⁴	8.75 (1.94) (55) ⁵	1.15 (0.37) (58)	6.05 (2.24) (67)

Note. Statistically significant results are printed bold

F = One-way Anova;

MSE = Mean Square Error value

ES = Effect Size ((mean₂ – mean₁) / standard deviation₁);

Eta² = Proportion of Explained Variance

¹ F(1,34) = 5.299, p = 0.028; MSE = 102.75; ES = 0.69; Eta² = 0.14

² F(1,34) = 5.414, p = 0.026; MSE = 41.56; ES = 0.63; Eta² = 0.14

³ F(1,34) = 4.158, p = 0.049; MSE = 12.94; ES = 0.84; Eta² = 0.11

⁴ F(1,38) = 7.399, p = 0.010; MSE = 11.37; ES = 0.89; Eta² = 0.16

⁵ F(1,38) = 5.699, p = 0.022; MSE = 5.07; ES = 0.67; Eta² = 0.13

For the construction of a mental model and identifying window elements and objects it was found that the users of the visual manual learned more than the users of a textual manual. In retrospect it can therefore be concluded that the study satisfies the four conditions for effective illustrations as proposed by Mayer and Gallini (1990, p. 716). There is explanative text (“the text must present a cause-and-effect system that allows for qualitative reasoning”); there are explanative illustrations (“illustrations must help the learner build a runnable mental model of the system”), there are sensitive tests (“the performance measure must evaluate the learner’s understanding and qualitative reasoning about the system”), and the study involves inexperienced learners who need support (“the students must not spontaneously engage in active learning processes such as the construction of a runnable mental model of the system”). In this study the explanative illustrations are screen captures whose design is based on a framework of roles and designs. The presence of a statistically significant difference on the test scores for Chapter 3, and not the Chapters 1 and 2, seem surprising in this respect (see Table 5-8). Chapter 3 of the visual manual was meant primarily to support verification. The explanation may be that the Chapters 1

and 2 of the visual manual may have had a positive effect on the way users processed the information in Chapter 3. In other words, there may have been a delay in the creation of a mental model. Over time, the added effects of the visual manual become visible. Interestingly this effect co-occurs with a delay in the processing of the manual itself as there was a one week gap between processing the first two chapters of the manual and the third chapter. Although the specific effects of the illustrations could not always be proven directly, they did yield their intended effect albeit delayed. This nicely accords with the observation that a positive effect of pictures in instruction may grow over time (Peeck, 1974).

The results for training time clearly show that *locating window elements and objects* is facilitated by the visual manual. The positive outcome for training time of the visual manual seems to be primarily a factor of support for the process of 'locating' because only in the Chapters 1 and 2 the difference was statistically significant. In these chapters, the screen captures were specifically meant to support locating window elements and objects, which was not the case in Chapter 3 (see Table 5-5). Screen captures for verification should also ease task execution. However, this was not found for Chapter 3 in which the screen captures were designed to support exactly this purpose. There are two reasons that may explain why no effect on training time was found for this chapter.

First, the number of screen captures may be an explaining factor. Chapter 1 with 22 pages in the visual manual had an average of 2.36 pictures per page (the textual manual consisted of 13 pages). Chapter 2 of the visual manual consisted of 39 pages with an average of 3.41 per page (the textual manual consisted of 22 pages). Compared to these chapters, Chapter 3 contained relatively few screen captures. Namely, 0.88 pictures per page on a total of 35 pages (the textual manual consisted of 31 pages). The sheer abundance of pictures in the Chapters 1 and 2 may have been beneficial for learning. Indeed, it more than offset the presence of more than twice as many pages in the visual manual. In Chapter 3, the presence is not as dominant, which may perhaps support less often a multimodal form of encoding.

Second, the substantial difference between locating and verifying may also account for the finding. The function locating refers to the process of finding a relevant screen object or element as quickly as possible. The function verifying, in contrast, refers to a process of carefully checking the screen for possible errors. When verifying information is clear and easy to use, users are likely to use it as such and take time to verify. In such a case they may not speed up their process of verification but rather take a good deal of time in verifying whether their actual screen state is correct. When verifying information is less clear and easy to use, users are more likely to skip it and hence gain training time. This process is more likely to occur in the textual manual.

In contrast to the clear difference on the speed of locating, no differences were found on the ability to correctly locate objects en elements on the post-test. These findings seem to be inconsistent, but the explanation is quite straightforward. Both the visual and the textual manual contained extensive support for locating the relevant part of the screen. For the textual manual the information in the screen capture that was relevant for locating was 'translated' into text as good as possible.

This translation was successful, as shown in the finding that both manuals were equally supportive in correctly locating screen elements. However, the texts could not match the pictures for speed of locating these elements.

Screen captures were not found to support *verification*. This may have been an effect of the difficult process of measuring verification time. The logged time intervals were not detailed enough to measure the quick nature of the user's actions. In addition, time may also not be the most appropriate way to measure this function because the main goal in verifying is to do a good job instead of a quick one. Our effort to capture the verification process by inserting a few deliberate errors unfortunately failed due to ceiling effects. Clearly, the deliberate errors were too easy to overcome. The test thus lacked the sensitivity that is conditional for finding positive effects for effective illustrations (Mayer & Gallini, 1990). As a consequence, the benefits of presenting visual information to support verification processes could not be demonstrated.

The task used in this experiment differed from those in Mayer's experiments in the larger number of information sources (manual and computer screen) and input devices (keyboard and mouse) to be used at the same time, and the required interaction between and with these sources and devices. From the findings on learning effects and training time it can be concluded that also for this realistic, 'on the job' setting, the multimodal instruction has proven to be superior over a unimodal one. As such, the present study extends the existing findings in support of Dual Coding theory.

Finally, no drawbacks of using a visual design were found for cognitive load caused by redundancy. The scores for cognitive load are neither low nor high. They are at a fairly optimal level of moderate cognitive load. A low cognitive load will result in boredom, whereas (too) high cognitive load will frustrate learning. The results contradict the predictions that Cognitive Load theory gives for the effects of introducing visual information in a context as the present one in which the user sees pictures on the computer as well as in the manual. The expected extraneous cognitive load caused by this redundancy did not occur. The visual design of the instruction lead to improved learning, despite the redundant character of the instructional setting. We ascribe this abrogation of extraneous cognitive load to the functional use of screen captures. That is, the design of the instruction may have caused a decrease of extraneous cognitive load, possibly neutralizing extraneous load caused by redundancy effects. Dual Coding theory gives a valid explanation for the way combinations of text and pictures are processed in this study as shown by the decrease of training time and the increase of learning effects found for the visual manual.

In conclusion, this study shows that also in a realistic context, multimodal instruction leads to better outcomes than unimodal instruction. Training time is shortened, there is better learning, while at the same time cognitive load is not altered. In addition to Mayer's integrated design criterion for the use of visuals in instruction, the specific

functional utilization of screen captures seems conditional to entail this yield. Further research is needed to gain more detailed insight in the processing of information in complex multimodal learning. Dealing with several interacting information sources and input devices simultaneously and its consequences on processing in memory may go beyond the current explanations given by Dual Coding theory and Cognitive Load theory. Also, further exploration of screen capture roles is needed to provide for subsequent strengthening of multimodal instruction in realistic contexts.

References

- Chandler, P., & Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognition and Instruction*, 8, 293-332.
- Cohen, J. (1962). The statistical power of abnormal-social psychological research: A review. *Journal of Abnormal and Social Psychology*, 65, 145-153.
- Jih, H. J., & Reeves, C. T. (1992). Mental models: A research focus on interactive learning systems. *Educational Technology Research and Development*, 40, 39-54.
- de Jong, T., Van Joolingen, W. R., Swaak, J., Veermans, K., Limbach, R., King, S., & Gureghian, D. (1998). Self directed learning in simulation-based discovery environments. *Journal of Computer Assisted Learning*, 14, 235-246.
- Gellevij, M., Meij, H. van der, Jong, T. de, & Pieters, J. (1999). The effects of screen captures in manuals: A textual and two visual manuals compared. *IEEE Transaction on Professional Communication*, 42, 77-91.
- Gentner, D., & Stevens, A. L. (1983). *Mental models*. Hillsdale, NJ: Lawrence Erlbaum.
- Kulhavy, R. W., Stock, W. A., Peterson, S. E., Pridemore, D. R., & Klein, J. D. (1992). Using maps to retrieve text: a test of conjoint retention. *Contemporary Educational Psychology*, 17, 56-70.
- Mayer, R. E. (1999). Designing instruction for constructivist learning. In: Reigeluth, C.M., *Instructional-Design theories and models Volume II: A new paradigm of instructional theory*. Mahwah, New Jersey: Lawrence Erlbaum Associates.
- Mayer, R. E., & Gallini, J. K. (1990). When is an illustration worth ten thousand words? *Journal of Educational Psychology*, 82, 715-726.
- Mayer, R. E., & Moreno, R. (1998). A Split-Attention effect in multimedia learning: evidence for dual processing systems in working memory. *Journal of Educational Psychology*, 90, 312-320.

Mayer, R. E., Moreno, R., Boir, M., & Vagge, S. (1999). Maximizing constructivist learning from multimedia communications by minimizing cognitive load. *Journal of Educational Psychology, 91*, 638-643.

Mayer, R. E., & Sims, V. K. (1994). For whom is a picture worth a thousand words? Extensions of a Dual-Coding theory of multimedia learning. *Journal of Educational Psychology, 86*, 389-401.

Murphy, C. A., Coover, D., & Owen, S. V. (1989). Development and validation of the computer self-efficacy scale. *Educational and Psychological Measurement, 49*, 893-899.

Norman, D. A. (1983). Some observations on mental models. In Gentner, D., & Stevens, A. L. (Eds.), *Mental models*. Hillsdale, NJ: Lawrence Erlbaum.

Nowaczyk, R. H., & James, E. C. (1993). Applying minimal manual principles for documentation of graphical user interfaces. *Journal of Technical Writing and Communication, 23*, 379-388.

Paas, F. G. W. C., Van Merriënboer, J. J. G., & Adam, J. J. (1994). Measurement of cognitive load in instructional research. *Perceptual and Motor Skills, 79*, 419-430.

Paivio, A. (1990). *Mental representations: a dual coding approach*. New York: Oxford University Press.

Peeck, J. (1974). Retention of pictorial and verbal content of a text with illustrations. *Journal of Educational Psychology, 66*, 880-888.

Peeck, J. (1993). Increasing picture effects in learning from illustrated texts. *Learning and Instruction, 3*, 227-238.

Robinson, D. H., Robinson, S. L., & Katayama, A. D. (1999). When words are represented in memory like pictures: evidence for spatial encoding of study materials. *Contemporary Educational Psychology, 24*, 38-54.

Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and instruction, 4*, 295-312.

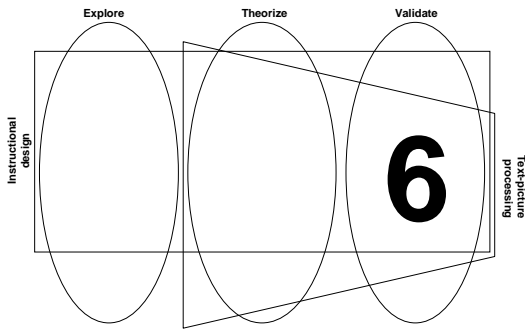
Sweller, J., & Chandler, P. (1994). Why some material is difficult to learn. *Cognition and Instruction, 12*, 185-233.

Van der Meij, H. (1996). A closer look at visual manuals. *Journal of Technical Writing and Communication, 26*, 371-383.

Van der Meij, H. (2000). The role and design of screen images in software documentation. *Journal of Computer Assisted learning*, 16, 294-306.

Van der Meij, H., & Gellevis, M. (1998). Screen Captures in software documentation. *Technical Communication*, 45, 529-543.

Veer, G. V. van der (1989). Individual differences and the user interface. *Ergonomics*, 32, 1431-1449.



Validation: Verification of Screen States

Abstract – This study set out to validate the supportive role of screen captures for the verification of screen states. Fifty-three participants learned how to work with a complex software application with a paper manual through self-study. There were two types of manuals: a textual manual and a visual manual in which screen captures were used to optimally support the verification process. During the 18-23 minute training sessions, time, ability to recover from errors, and cognitive load were measured. The findings show that participants of the visual manual were significantly better in verifying. That is, they were quicker in accomplishing verification tasks and made significantly fewer mistakes. Cognitive load was equal over conditions on both a self-scoring and a secondary task measure. After training, participants were asked about their opinions about screen captures by manuals. Here, a difference was found between expected value of screen captures by users of the textual manual, which was significantly lower than experienced value of screen captures by users of the visual manual.

Introduction

Many people still see a computer as a delicate instrument that can easily be broken. Especially novice users may fear that their actions will damage the computer or the application. Their attitude towards working with computer may therefore initially be rather reserved. Research corroborates this impression. As people become more experienced, their attitude towards the computer changes and their self-confidence increases (Gardner, Dukes, & Discenza, 1993; Pope-Davis & Twing, 1991). The manual can alleviate initial fears by presenting information that allows users to verify that they are on the right track. For example, after pressing a key, novices want to know whether they have done the right thing. A potentially strong form to present such verification information is screen captures. Screen captures make it possible to verify screen states and, thus, confirm or contradict the user's progress (Horton, 1993; Price & Korman, 1993).

Apart from providing positive feedback, screen captures can help users detect errors early on and thus facilitate error management. Screen captures are supposed to be optimally suited for this verification process. They make it relatively easy to compare the information in the manual with the actual computer screen. In contrast to using text, the user does not need to process or code the textual information into a visual; comparing a picture of the screen with the actual screen is easier than comparing the content of written text with a screen.

Chapter 3 (see also: Van der Meij and Gellevij, 1998) provided a taxonomy of four roles or functions of screen captures in manuals. So far, three of these functions have been validated. Screen captures have been found to support users to *switch attention* between the manual and the screen (see Chapter 4; Gellevij & Van der Meij, submitted), they support the construction of a *mental model* of the computer program, and they help to *identify and locate* screen elements and objects (see Chapter 5; Gellevij, Van der Meij, De Jong, & Pieters, in press). The purpose of this paper is to validate the fourth function: *verification of screen states*.

Price (1984) advises to include pictures of the screen in tutorials to let users know that they have got the right display and have done the right thing. From this, two types of verification can be distinguished: verification of program-driven progress, and verification of task-driven progress.

Program driven verification is determined by the way the program is setup. To be able to work with the program, one should follow a series of steps like using menu-options, opening and closing certain windows, etcetera. Verification of program-driven progress is checking whether the correct menu or window indeed (dis)appears.

Verification of task-driven progress deals with the verification of task specific actions. It is about what a user actually tries to accomplish with the computer program; about the assignment he or she has to carry out. In practice this means verifying what someone does within a certain window. Both types of verification are

supported in this study although the distinction between the two is not further studied.

In an earlier study we already tested for effects of using screen captures for verification purposes (see Chapter 5; Gellevij et al., in press). Ease of verification, measured by time on verification tasks, and quality of verification, measured by the ability to recover from errors did not result in evidence to validate this function due to methodological issues. The logged time intervals were not detailed enough to measure the specific verification parts of the task, and the errors were too easy to overcome which caused a ceiling effect. The current study can be considered a replication that attempts to remedy these methodological issues. That is, ease and quality of verification by using screen captures remain to be the primary focus of this study.

Much related to these *outcomes* of verifying screen states is the way how screen captures for verification are being processed in memory *during* task execution. It is important to investigate how screen captures influence the load imposed on working memory during task execution. Our interest in cognitive load when processing text-picture combinations comes from two seemingly conflicting theories on this issue. Dual Coding theory argues that working memory consists of two distinct systems: a verbal and non-verbal system. Using the capacity of both systems leads to more information being processed than when using only one of the systems. In addition, it also yields better results because the simultaneous processing renders the connectivity of the two systems. This referential connectivity in turn contributes to the construction of a strong mental model (Paivio, 1990; Mayer, 1999). Thus, using text and screen captures in a manual will improve learning because working memory is optimally used.

Cognitive load theory argues that a visual manual causes too much cognitive load because of the redundant character of the screen captures. The redundancy hypothesis predicts that offering the same information twice, as in the case of a depicted computer screen, requires the user to process this information twice. The double processing takes up unnecessary memory space, space that could have been used to process other information. According to Sweller and Chandler (1994), redundant information is therefore likely to increase training time and decreases learning effects because memory is *not* optimally used.

In a previous study (Gellevij et al., in press), we measured cognitive load and found no differences between users of a textual and visual (text plus screen captures) manual. Cognitive load did not increase when using screen captures in manuals, contrary to predictions by Cognitive Load theory. This finding, in combination with the increase on learning and reduced training time for users of the visual manual are favoring the Dual Coding explanation that screen captures are beneficial rather than obstructive. In this study we again measure cognitive load to check if this conclusion holds.

Cognitive load is usually measured by self-scoring (on a nine-point Likert scale), using questions like: 'How difficult would you rate the task at this moment?'. Paas,

Van Merriënboer, and Adam (1994) have shown this to be a reliable and more sensitive way to measure cognitive load than an objective measurement as heart rate variability. As an alternative way of measuring cognitive load, we take the middle position between subjective and objective testing, namely measuring response times by letting a user shut off a sound. We argue that the time a user takes to get distracted from his or her its primary task can serve as a good indication for his or her cognitive load.

In this study, a textual manual is compared with a visual manual in which screen captures are used that explicitly support the function of verification. Verification entails checking progress and recovering from errors. The prediction is that users of the visual manual show a higher quality of verification than users of the textual manual. In short, we predict that users of the visual manual will make fewer mistakes.

We also predict that cognitive load will be equal for both conditions, contradicting the hypothesis given by Cognitive Load theory that predicts that cognitive load will be higher in the visual condition. In addition, we test whether a secondary task measure on response times is a valid alternative way of measuring cognitive load.

Finally, it is hard to give a prediction for ease of verification. The function verifying refers to a process of carefully checking the screen for possible errors. Such a process may take time. Therefore, a trade-off between time and error-recovery may be expected, meaning that high quality of verification will lead to long verification time, and vice versa. However, a similar trade-off was expected in an earlier study but then between time and learning effects (Gellevij, Van der Meij, de Jong, & Pieters, 1999). That study showed that such a trade-off does not necessarily exists. As a time gain and improved learning can go hand-in-hand, so can a time gain and quality of verification.

Method

Participants

Fifty-three international students (20 males and 34 females) from the Faculty of Educational Science and Technology participated in the study. This user group was selected because of their rather high level of computer knowledge and skills. The participants had no experience with the computer program used in the experiment.

Materials

Computers

The sessions were held with use of two IBM compatible computers. One computer was used for the main task: solving problems in SimQuest (see next section) based on the information in the paper manual. Participants carried out this task using the keyboard, mouse, and computer screen. The other computer was used to measure

cognitive load. Participants only used the mouse of this computer to turn off a sound that was produced by this computer. The keyboard and screen were not accessible.

SimQuest

The computer program SimQuest (van Joolingen, & de Jong, in press; de Jong, et. al., 1998) consists of two parts: a learner part and an authoring part. SimQuest applications offer learners discovery learning environments in which they can explore a specific domain. Their discovery behavior is evoked through simulations. To support the discovery process, learners receive assignments and explanations in forms such as video, (html-)texts, pictures, and sound.

Teachers or instructional designers use the SimQuest authoring environment to modify an existing learner environment or create a new one. SimQuest is based on an object-oriented approach, which means that a collection of ready-made elements can be used to create an application or program. In the reported study, participants solved problems and removed errors in the SimQuest authoring environment.

Manuals

The two manuals (textual and visual), consisted of an introductory chapter on organizing SimQuest, and five short chapters about various changes that can be made in the SimQuest authoring environment such as changing an interface, interface elements, assignment, explanations, and contexts. The only difference between the textual and visual manual was the absence of screen captures in the textual manual. The lay-out and typography of the visual and textual manual were further as similar as possible to avoid differences in reading and to make sure that effects could be attributed to the screen captures.

Figure 6-1 shows the screen capture design used in the visual manual. The design arguments that led to this design, were: (1) providing a size that is legible, to make sure that the specific content can easily be verified, (2) showing only the relevant part of the screen to help focus attention, and, (3) if necessary, using a cueing technique to point to the essential verification part.

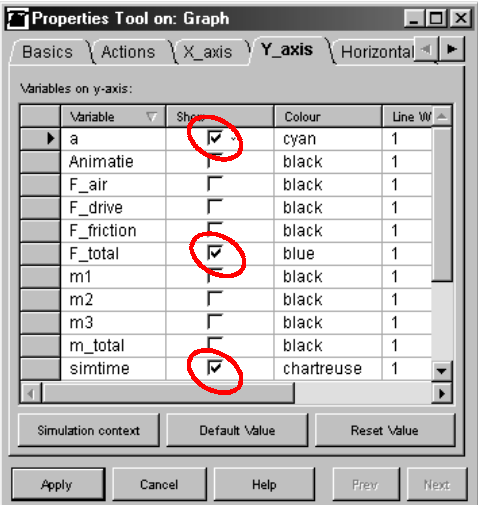
The presence of screen captures led to manuals of different sizes, which may influence the speed in which the manual can be processed. That is, the more pages, the more time someone may need to work through that manual. Table 6-1 shows the number of pages for each chapter of the manuals and the number of screen captures used.

Table 6-1. Number of manual pages and number of screen captures in the textual and visual manual

Chapter	Pages		Screen captures	
	Visual	Textual	Visual	Textual
Organizing the system	2	1	2	0
Changing an Interface	4	2	6	0
Changing a context	4	2	7	0
Changing interface elements	4	2	7	0
Changing assignments	3	1	5	0
Changing explanations	3	1	6	0
Total	20	9	33	0

Visual manual

1 Check, in the *Properties Tool on: Graph*, if the boxes in the *show* column of the variables **a** , **F_total**, and **simtime** are marked



If not, mark the appropriate boxes by clicking in them

Textual manual

1 Check, in the *Properties Tool on: Graph*, if the boxes in the *show* column of the variables **a** , **F_total**, and **simtime** are marked

If not, mark the appropriate boxes by clicking in them

Figure 6-1. Screen capture design for the function *verifying screen states*, and the design of the textual counterpart

Questionnaires and tests

Before the experimental task, participants were asked about their personal characteristics like *gender* and *age*. They were also asked to grade themselves, on a scale from 1 (low) to 10 (high) on *computer experience*. They were explicitly told that this did not mean ‘programming’, but day-tot-day tasks, for example using Microsoft Office applications, reading email, etcetera.

During the experimental task, the observer wrote down the *time* for certain action steps, and scored the *errors* that were made. The manual contained intentional errors, to afford for error verification. In these cases, the information that the user must verify in the screen capture, or text, deliberately differs from the correct situation at that moment. An example of an intentional error is shown in Figure 6-1. On the actual screen, the boxes are *not* marked. Based on the verification information the user must mark the appropriate boxes. The software program Camtesia™ Recorder was used to record the users’ actions on the screen. The films that were obtained this way served as a check to provide for reliability of the observations. Observations and films of three randomly selected participants were used to calculate the reliability. Cohen’s kappa was 1 for the amount of errors made. Observed time and logged time differed on an average of 3.46 seconds ($sd = 2.45$)¹.

Participants were to shut off a sound produced by a second computer as quickly as possible to measure their *response time*, which was used as a measurement for cognitive load. The sound was produced once every two minutes. Pilot tests showed that such a time interval did not lead to crucial disturbance of the primary task. After every chapter in the manual, participants were asked: ‘How would you rate the difficulty of this task?’ They indicated their *task-difficulty* by putting a mark on a 10 centimeter long line that represented the continuum from ‘very easy’ to ‘very difficult’. There were six of such measurements, which served as a second measurement for cognitive load.

After the experimental task, participants were asked to give their *opinions* about the manual that they had worked with (see Table 6-2). Participants answered these questions the same way as with task difficulty, by marking a 10 centimeter long line.

¹ This difference can be explained by the fact that the observer wrote down the time when a participant started to read an action step. The logged time is the mouse click of that action step, which is a few seconds later.

Table 6-2. Questions about the manual (range 0 – 10)

	Answer (range 0 – 10)
G1. How clear was the text for you?	Not – Very
T1. Would you liked to have screen pictures in the manual?	None – A lot
T2. Would screen pictures help you with the task?	Not – A lot
V1. Did you use the screen pictures in the manual?	Not – Always
V2. How useful were the screen pictures for you?	Not – Very
V3. If there were no screen pictures, how would the difficulty of the task change for you?	Less difficult – More difficult

G = General question asked to both groups; T = Asked to Textual group; V = Asked to Visual group

Procedure

The experiment was held using one-on-one observations. Participants were told that the goal of the study was to investigate the way people behave while they individually learn to work with a computer program, using a paper manual as the only source of information. A participant was seated behind the computer and was asked to think-aloud while working with the manual and the computer. The observer took place at the left-hand side of the participant sitting in a 45° angle, so he could see the computer screen as well as the hands of the participant to be able to observe the participants actions. A 45° angle was chosen over a 90° angle to prevent the participant to get distracted by the observer.

During the experimental session, the observer wrote down the start- and end-times of verification operations and scored the results of the problems solved and errors removed. This session took about half an hour. During the session, participants shut off the sound and rated their task-difficulty. After the session, participants filled in the opinion-questionnaire, and were finally informed about the setup and goals of the experiment.

Coding and scoring

Number of participants in the analyses

Fifty-three students participated in the study, who were in general all included in the analyses. Because of the speed of the users’ actions, the observer sometimes was not able to record some data, which is why there are minor variations in the number of participants in the various analyses.

The average score on response times of one participant was excluded from the dataset because his data was considered unreliable. The participant did not understand the instruction to turn off the sound as quickly as possible and had to be prompted to do so. This resulted in an average response time that was almost six times above average.

Gender, age, and computer experience

Data on gender, age, and computer experience did not need to be (re)coded. They were used in the analyses in their original form.

Time

The first part of the manual was treated as a practice task. Time used to carry out the total task was therefore calculated as the end time minus the time after accomplishing the practice task.

Furthermore, time was calculated for (18) specific verification tasks. Time used on these tasks was calculated by subtracting the time of the action step before verification from the time of the action step after verification.

Progress checks and error recovery

The participants could make two types of errors: (1) leave things unchanged that were required to be changed, and (2) changing things that did not require to be changed according to the information in the manual. Both types of errors were scored as one point for each occurrence.

Cognitive load

Cognitive load was measured in two ways, by measuring response times and by self-scoring. Because of the variation in training time, the number of response times measured also varied (range 8-25). The first three measurements were excluded because they were taken during the practice task. The remaining response times were recalculated into an average. A participant's average response time was used in the analyses.

Cognitive load and the questions on users' opinions about the manual were measured by self-scoring. Participants answered these questions by putting a mark on a 10 centimeter long line. Scores on cognitive load and opinions were obtained by determining the place of the mark with use of a ruler, resulting in scores ranging from 0 to 10 with one decimal (e.g. 5.4). Pearson correlations revealed strong and statistically significant coherence between the six measured moments of the self-scored cognitive load (see Table 6-3). Therefore, an average was calculated for these measurements and this average was used in the analyses.

Table 6-3 Correlations between self-scored cognitive load measures (n=54)

	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5
Chapter 2	0.429				
Chapter 3	0.567	0.617			
Chapter 4	0.445	0.517	0.737		
Chapter 5	0.455	0.456	0.594	0.700	
Chapter 6	0.426	0.505	0.706	0.574	0.742

Note. All correlations are statistically significant at the 0.01 level (1-tailed)

Opinions

Strong and statistically significant (2-tailed) Pearson correlations (0.926, $p = 0.000$) were found between the two opinion questions about screen captures that were asked to the users of the textual manual (see Table 6-2, questions T1 and T2), indicating that these questions measured the same. The same holds for the two opinion questions that were asked to the visual group (see Table 6-2, questions V2, and V3) where the correlation was 0.506 with a significance level of 0.008. Based on these correlations, the two pairs of questions have been combined into one for each group, giving a general indication on how users value screen captures in software manuals. Results on the question ‘Did you use the screen pictures in the manual’ (see Table 6-2, question V1) were not used in the recalculation, as this question did not ask for an opinion, but was meant to check perception of actual use.

Results

Check for randomization

To check the random distribution of participants over the three conditions, Anova’s were used comparing the conditions on the covariates computer experience and age. A Chi-square test was used on the covariate gender. Results of these analyses showed a random distribution on all three variables (see also Table 6-4). Means (standard deviations) for age in the textual and visual condition were 29.44 (6.78) and 28.08 (3.72) years. The means (standard deviations) for computer experience were 7.61 (1.15) and 7.48 (1.00) respectively, on a scale from 1 to 10.

Table 6-4. Crosstabs for gender by condition

	Male	Female	Total
Textual	7	20	27
Visual	12	14	26
Total	19	34	53

Time and problem solving

Main effects were found for the time used for the total task and the amount of errors made (see Table 6-5). Users of the visual manual used 22% less time and made 2.69 times less errors than users of the textual manual. Both differences are statistically significant.

Table 6-5. Means (standard deviations) on total time in minutes and amount of errors

	Textual	Visual
Time (minutes)	23.15 (6.30) (n=22)	18.16 (4.05) (n=26) ¹
Errors	3.63 (3.04) (n=27)	1.35 (1.72) (n=26) ²
Verifying (4 tasks)	0.33 (0.55) (n=27)	0.04 (0.20) (n=26) ³
Verifying and changing (14 tasks)	3.30 (2.91) (n=27)	1.31 (1.62) (n=26) ⁴

F = One-way Anova;

ES = Effect Size $((\text{mean}_2 - \text{mean}_1) / \text{standard deviation}_1)$;

Eta^2 = Proportion of Explained Variance

¹ $F(1,46) = 10.386$, $p = 0.002$; ES = 0.79; $\text{Eta}^2 = 0.18$

² $F(1,51) = 11.211$, $p = 0.002$; ES = 0.75; $\text{Eta}^2 = 0.18$

³ $F(1,51) = 6.554$, $p = 0.013$; ES = 0.53; $\text{Eta}^2 = 0.11$

⁴ $F(1,51) = 9.368$, $p = 0.004$; ES = 0.68; $\text{Eta}^2 = 0.16$

Table 6-6 shows the time spent on verification tasks. There were 18 of such tasks. In four of these, the information in the manual was the same as that presented on the screen. In these cases, the user just had to verify this similarity. In the remaining 14 verification tasks, information in the manual and on the screen differed, requiring some kind of action of the user to make a change. As Table 6-6 shows, time used on these two types of tasks differed statistically significant between conditions.

Figure 6-2 gives graphical impression on the time used for the 18 verification tasks for both conditions. The figure shows an equal pattern for both groups, but time is almost consistently lower for the visual condition. Multivariate Repeated Measures analyses points to an interaction between time and condition ($F(17,12) = 3.822$, $p = 0.011$). Users of the visual manual generally are faster in completing verification tasks than users of the textual manual.

When making the distinction between verification tasks that do *not* require a change (see Figure 6-2, tasks 1, 2, 6, and 7) and verification task that do need a change, Multivariate Repeated Measures analyses points to an interaction between time on verification and condition ($F(3,40) = 3.373$, $p = 0.028$). Users of the visual manual generally are faster in just verifying than users of the textual manual. This interaction does not reach level of significance for verification tasks were a change is required ($F(13,21) = 1.991$, $p = 0.077$)

Pearson Correlation between time and amount of errors is 0.363 ($n = 48$) with a significance level of $p = 0.011$, meaning that participants who took more time also made more errors. No statistical significance is found when these correlations are calculated for the separate conditions.

Table 6-6. Means (standard deviations) on verification time in seconds and amount of errors

	Time		Errors	
	Textual	Visual	Textual	Visual
Total (18 tasks)	756 (136) (n=14)	604 (161) (n=16) ¹	3.36 (3.00) (n=14)	1.19 (1.68) (n=16) ⁴
Verifying (4 tasks)	182 (83) (n=20)	123 (52) (n=24) ²	0.35 (0.59) (n=20)	0.04 (0.18) (n=24) ⁵
Verifying and changing (14 tasks)	589 (102) (n=19)	474 (129) (n=16) ³	3.11 (2.69) (n=19)	1.13 (1.50) (n=16) ⁶

F = One-way Anova;

ES = Effect Size ((mean₂ – mean₁) / standard deviation₁);

Eta² = Proportion of Explained Variance

¹ F(1,28) = 7.648, p = 0.010; ES = 1.12; Eta² = 0.21

² F(1,42) = 8.395, p = 0.006; ES = 0.71; Eta² = 0.17

³ F(1,33) = 8.712, p = 0.006; ES = 1.13; Eta² = 0.21

⁴ F(1,28) = 6.164, p = 0.019; ES = 0.72; Eta² = 0.18

⁵ F(1,42) = 5.801, p = 0.020; ES = 0.53; Eta² = 0.12

⁶ F(1,33) = 6.873, p = 0.013; ES = 0.74; Eta² = 0.17

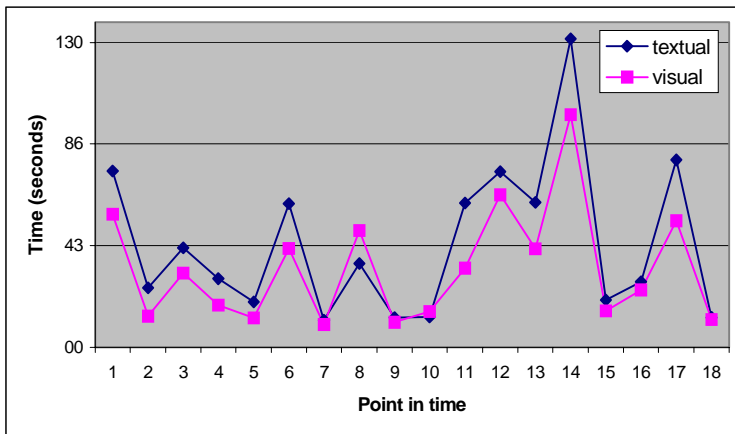


Figure 6-2 Time used on the (18) verification tasks

Cognitive load

No differences were found on cognitive load. Both the average response time and the average score on task difficulty were equal over conditions (see Table 6-7). With p = 0.054, Pearson Correlation of 0.225 (n = 52) between both cognitive load measurements almost reaches a (1-tailed) level of significance. This correlation is

statistically significant for users of the textual manual (0.382, $p = 0.027$, $n = 26$) but not for users of the visual manual (0.097, $p = 0.319$, $n = 26$).

Table 6-7. Means (standard deviations) on response time and cognitive load self-scoring

	Response time (milliseconds)	Task difficulty (range 0-10)
Textual	2095 (418) ($n=26$)	3.31 (1.66) ($n=27$)
Visual	2045 (517) ($n=26$)	3.10 (1.68) ($n=26$)

Figure 6-3 shows the cognitive load scores for both conditions based on self-scoring. The figure shows a comparable pattern for both groups. Multivariate Repeated Measures analyses did not point to an interaction between cognitive load and condition ($F(5,47) = 2.386$, $p = 0.052$). Cognitive load can therefore be considered the same for both users of the textual manual and the visual manual.

Statistically significant Pearson correlations are also found between the cognitive load measurements and the time and error measurements. Table 6-8 shows that in all cases, cognitive load correlates positively with time and amount of errors, meaning that the more taxing the task was for working memory, the more time users needed to accomplish it, and the more errors they made, and vice versa, the less load on working memory, the fewer time was needed and fewer errors were made.

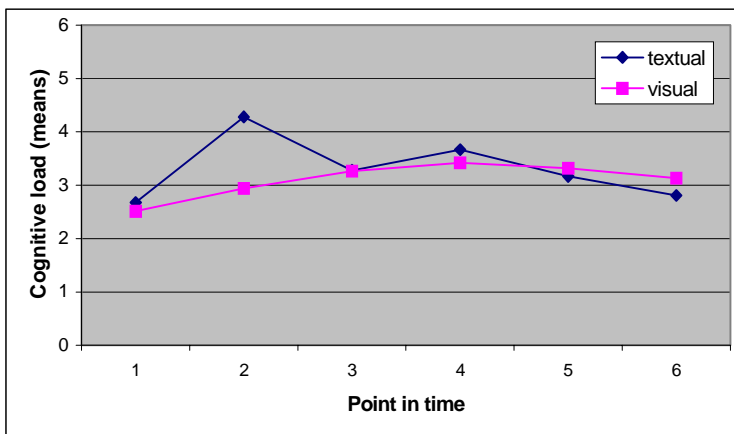


Figure 6-3 Cognitive load in the course of time based on self-scoring

Table 6-8. Correlations between cognitive load, time and errors

	Response time (milliseconds)	Task difficulty (range 0-10)
Time	0.397 (n=47) **	0.346 (n=48) *
Errors	0.346 (n=52) *	0.383 (n=53) **

* $p < 0.05$, ** $p < 0.01$ (2-tailed)

Opinions

After the experimental task, the participants were questioned about the manual they just used. Both groups were asked how they perceived the clarity of the text. Users of the textual and the visual manual graded text-difficulty equally. The mean scores (standard deviations) for the textual and visual users were 7.08 (2.15) and 6.70 (2.39) respectively.

To get an impression whether users of the visual manual actually used the screen captures the question ‘Did you use the screen pictures in the manual?’ was asked. On an scale from 0 to 10, users of the visual manual scored a mean (standard deviation) of 5.90 (2.69), indicating that screen captures were indeed perceived as being used.

Based on a combination of questions as presented in Table 6-2, an opinion on how screen captures were, or would be, valued was obtained. The difference between these value scores, 5.68 (2.62) for textual and 7.37 (1.85) for visual users, is statistically significant ($F(1,51) = 7.335$, $p = 0.009$). Comparing these value measurements shows that appreciation of the screen captures by users of the visual manual, is significantly higher than the expected appreciation of screen captures from the users of the textual manual.

Discussion

Practitioners have pointed out that verification of screen states is probably an important advantage of the display of screen captures (Horton, 1993; Price, 1984). The findings in the current study confirm this view. Ease of verifying and quality of verification are higher when users are supported by screen captures in a software manual. Evidence to validate the verification function of screen captures is further strengthened by the fact that the potential trade-off between time and the effectiveness of problem-solving does not exist. Instead of needing a lot of time to carefully recover from errors, users of the visual manual were not only better in solving problems than users of the textual manual, they also did it in a remarkably shorter period of time even despite the fact that they had to work through more than double the amount of pages. Visual users used 22% less time on their task than textual users and made more than 2.5 times fewer errors. Effect sizes of these findings are around or well above 0.70. As an effect size of 0.25 is generally seen as a small effect, 0.50 as a medium sized effect, and 0.75 as a large effect (Cohen, 1962), the effect sizes for ease and quality of verification can be considered quite high. These findings indicate that, after having overcome some methodological obstacles, research clearly validates the verification function of screen captures.

The findings on cognitive load are similar to our earlier examination. Again we found no increase in cognitive load due to using a visual design. In combination with the results on time and error-recovery, it may be concluded that there is no negative redundancy of screen captures as suggested by Cognitive Load theory. It seems clear that although the screen and its picture in the manual show basically the same information, the function of both information sources is different, giving either one a distinct role. Instead of increasing cognitive load that bothers optimal use of working memory, the screen captures serve essential functions. In this respect, the study supports the views of Dual Coding theory.

Apart from self-scoring, we used a secondary task measure to measure cognitive load. Our starting-point was that the time needed to get distracted from the primary task would give a good indication for cognitive load. The findings on this secondary task measure are comparable to the traditional scoring, but our findings are not conclusive on the interchangeability of both measurements. The correlation is rather low and does not reach a desirable level of statistical significance. The experimental setup also makes it questionable whether both measurements can be seen as strictly independent when used at the same time. For example, when users notice that they were rather late in shutting off the sound, they may have felt 'obliged' to rate the task as rather difficult. The findings on the two measurements of cognitive load thus only give an indication that both ways of testing measure the same.

Finally, we found that users need to see and experience something first, before they actually believe it. The participant's opinions about the value of screen captures in manuals, show that users value screen captures significantly higher when they have actually been using a visual manual than users of a textual manual do when asked about the potential value of screen captures in a software manual. It seems clear here that 'the proof of the pudding is in the eating'.

References

Cohen, J. (1962). The statistical power of abnormal-social psychological research: A review. *Journal of Abnormal and Social Psychology*, 65, 145-153.

Gardner, D. G., R. L. Dukes, & Discenza, R. (1993). Computer use, self-confidence, and attitudes: A causal analysis. *Computers in Human Behavior*, 9, 427-440.

Gellevij, M., Van der Meij, H., De Jong, T., & Pieters, J. (1999). The effects of screen captures in manuals: Textual and visual manuals compared. *IEEE Transactions on Professional Communication*, 42, 77 - 91.

Gellevij, M. R. M., Van der Meij, H., de Jong, T., & Pieters, J. M. (in press). Multimodal versus Unimodal Instruction in a Complex Learning Context. *Journal of Experimental Education*.

Gellevij, M. R. M., & Van der Meij, H. (submitted). Screen Captures to Support Switching Attention. *IEEE Transactions on Professional Communication*.

Horton, W. (1993). Dump the dumb screen dumps. *Technical Communication*, 40, 146-148.

Jong, T. de, Joolingen, W. R. van, Swaak, J., Veermans, K., Limbach, R., King, S., & Gureghian, D. (1998). Self directed learning in simulation-based discovery environments. *Journal of Computer Assisted Learning*, 14, 235-246.

van Joolingen, W. R., & de Jong, T. (in press). SimQuest: Authoring educational simulations. In T. Murray, S. Blessing & S. Ainsworth (Eds.) *Authoring tools for advanced technology educational software: Toward cost-effective production of adaptive, interactive, and intelligent educational software*. Kluwer Academic Publishers.

Mayer, R. E. (1999). Designing instruction for constructivist learning. In: Reigeluth, C.M., *Instructional-Design theories and models Volume II: A new paradigm of instructional theory*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Paas, F. G. W. C., Van Merriënboer, J. J.G., & Adam, J. J. (1994). Measurement of cognitive load in instructional research. *Perceptual and Motor Skills*, 79, 419-430.

Paivio, A. (1990). *Mental representations: a dual coding approach*. New York: Oxford University Press.

Pope-Davis, D. B., & Twing, J. S. (1991). The effects of age, gender, and experience on measures of attitude regarding computers. *Computers in Human Behavior*, 7, 333-339.

Price, J. (1984). *How to write a computer manual: A handbook of software documentation*. Menlo Park, CA: Benjamin/Cummings.

Price, J., & Korman, H. (1993). *How to communicate technical information*. Redwood City, CA: Benjamin/Cummings.

Sweller, J., & Chandler, P. (1994). Why some material is difficult to learn. *Cognition and instruction*, 12, 185-233.

Van der Meij, H. & Gellevij, M. R. M. (1998). Screen captures in software documentation. *Technical Communication*, 45, 529-543.

7

Conclusions and Discussion

This chapter reconsiders the theoretical approaches that were presented and used in this book in light of the findings of the studies performed. The chapter is organized according to the two main perspectives from which this book originated: instructional design and text-picture processing. The related goals of the reported studies were (1) to find ways to support learners in their process of learning by use of screen captures, and (2) to gain insight in the way combinations of text and pictures are processed in working memory. The chapter ends with suggestions for future studies on (a) the integration of the four functions of screen captures that in this work were studied separately, and (b) a more in-depth explanation for text-picture processing in working memory.

Instructional Design: Functions of screen capture functions

In this book, four functions of screen captures in software manuals have been presented. In general, the studies in this book show that functionally applied screen captures offer support for users who learn to work with a computer program based on a paper manual. The specific functions for screen captures that support the user's task execution were: switching attention between manual and computer screen, developing a mental model of the program, verifying screen states, and identifying and locating window elements and objects. All four screen capture functions have found to be supportive. To start with the verification function, the study presented in Chapter 6 shows that users are better in recovering from errors with visually supported verification information. It also shows that users need less time to carry out verification tasks than with only textual information. Proof was found for the construction of a stronger mental model of the program, better identification of window elements and objects, and faster localization by using screen captures in the study described in Chapter 5. As shown in Chapter 4, there is evidence as well for enhanced switching between manual and the computer screen, although the evidence for this function of screen captures is not entirely convincing.

During the implementation and examination of the four functions, views were sharpened, and new or further insights obtained. This section addresses the following ones. First, it is disputed that the validation studies in Chapters 4, 5 and 6 follow a genre-based approach as used in the exploratory study in Chapter 2. Second, the merit of the distinction between the four screen capture functions is explained. Third, it is discussed how the four functions of screen captures relate to each other.

Applying functions and a genre approach

In the explorative study presented in Chapter 2, visual manuals were used in which a genre-based based approach was adopted, meaning that throughout the manual, the same type of screen captures were presented regardless of the user's task. In this book, it was shown that a function-based approach, in which screen captures are deployed as support for specific user tasks, improves the instruction. Apart from the coupling of screen captures to user tasks, it was argued that the design of the screen capture should be maximally tuned to its function. For this, four design dimensions were given (see Appendix D) on which a grounded design for a function could be developed. In the absence of research on function-design relations, the screen captures used in the studies in this book were the expected best fit between the function of a screen capture and its design.

The screen capture designs used in the validation studies could lead to the impression that here too, a genre-based approach is followed, as the screen capture designs for the functions mental model development, identification and localization of screen elements and objects, and switching attention, are all rather similar. These screen captures thus serve more than one function. However, this resemblance in design between the function-based manuals and the genre-based manual does not make them identical. There are three arguments that corroborate such a view.

First, unlike in a genre-based approach, when taking a function-based approach the screen captures are only placed in the manual if there is a clear purpose for them. This means that in a function-based manual, sometimes parts of the instruction are intentionally *not* supported by a screen capture, whereas in a genre-based manual support by screen captures would be continuous.

Second, the function-based approach, unlike the design-based approach, deliberately uses different designs for different information types. Even within one function, different screen capture designs can be used, depending on the specific type of information that needs to be supported. For example, the construction of a mental model and the identification and localization of screen element and objects involve carrying out series of action steps and studying explanations, which are represented in the manual as procedural and conceptual information. These different information types require different screen capture designs, to maximize the support for learning.

Third, the screen capture designs vary as they are maximally tuned towards the function they intend to support. As indicated earlier, this may lead to similar designs for different functions. It may also lead to screen capture designs that are deliberately different over functions. This is illustrated by the design chosen for the function verifying screen states. The task that needs to be supported determines the design of

the screen capture. Compared to tasks that underlie the other screen capture functions, the verification task is quite different which requires a divergent and unique design.

Reasons for the distinction of screen capture functions

Throughout this book, functions are treated as being exclusive. The most important reason for this is the ability to examine them in their pure form, to obtain effects that can be ascribed to a specific function. That is, by isolating a function, one gets the opportunity to focus on only that function without being bothered by other, possibly disturbing, functions. Such an approach was considered necessary, as before the start of the studies, based on previous research, there were no indications about the benefits or drawbacks of specific screen capture functions. Testing functions in combination would therefore have been potentially troublesome, as the benefits of one function could have been cancelled out by drawbacks of another function. For example, it was quite difficult to predict if supporting the process of verifying by a screen capture would lead to a decrease or an increase on training time. After all, checking whether the information in the manual is exactly the same as on the screen may be a time-consuming process. Testing the function verification in combination with, for example, the function localization could result in findings that would have been hard to interpret. If no differences between a visual and textual manual on training time would have been found, the conclusion would have been that screen captures do not support the localization process. Findings on training time of the study in Chapter 5 show otherwise, namely, that screen captures do support the localization process. By studying the functions in isolation, the value of a distinct function is revealed.

This does not necessarily mean that the content of the manual in the experimental settings can only focus on the one user task a function is derived from. If that were the case, the manual would be quite unrealistic. In the studies of this book, the manuals contained a combination of tasks that usually comprised all four functions. Using screen captures in the visual manuals to support only one of those functions led to a realistic setting, in which a function could still be studied in isolation.

There can however be a reason to step down from the demand for a manual with realistic content based on a combination of user tasks. When the function under study is based on a task that occurs only every now and then, it may be necessary for research purposes to readjust the content of the manual so that the task is carried out more frequently than can be expected in reality. This was the case in the study to validate the function verification of screen states presented in Chapter 6. Verification was tested by letting users check their progress and recover from errors that were put in the manual deliberately. Such intentional errors are clearly unrealistic, but so was

the relatively large amount of progress checks. To have the same amount in a more realistic situation would have led to the need for too lengthy experimental sessions¹.

Relations between screen capture functions

The four screen capture functions are examined as distinct functions. This does not mean that these functions are not related. This section discusses how the screen capture functions depend on each other from the view that switching attention is of a different nature than the other functions.

In general, the screen captures functions all intend to support learning. Each function contributes to the learning process in a different way. The function mental model development supports the construction of knowledge about the way a program works and how it is structured. The function identification and localization supports knowledge construction on a more detailed level, for example the construction of knowledge about the place, meaning, and use of more specific parts of the program (e.g., buttons, icons). The function verification supports learning by confirming the user's progress while carrying out their task, and by assisting with the recovery from errors. As these three functions are closely linked to the task to be performed and knowledge to be acquired, these functions support the learning process in a direct manner.

The function switching attention also supports the learning process but in a more indirect way. The main goal of this function is to induce and facilitate interaction between the manual and the computer screen. Screen captures supporting switching attention are meant to provoke the user to look up from the manual towards the screen and back into the manual in such a comfortable way that the interaction takes place as an automated process. Switching behavior as such is not directly related to the learning task. But although support given for switching attention is indirect for learning, it cannot be seen as merely facilitative. In some cases, switching attention is even conditional for learning. In the training situations used in this book, where learning takes place while using the actual computer, comparing and finding specific information on the screen requires interaction between the manual and computer screen. Thus, the functions verifying and locating cannot be carried out properly without the accompanying function of switching attention. In other words, the function switching attention is conditional for the functions verification and localization.

The immediate question that arises from this assertion is whether switching attention is also conditional for the functions mental model development and identification. We argue that this is not the case. The construction of knowledge about (aspects of) the computer program can also take place based on the manual

¹ The argument is based on the assumption that the type of manual used is a tutorial. In tutorials, verification occurs regularly but not frequently. In other types of documentation, like for example job aids, users may need to engage in verification more often.

alone, without the interaction with the computer program. The studies in this book, however, show that realistic training situations in which such interactions do take place, lead to effective learning. While switching attention is not conditional for knowledge construction, the combination of functions certainly seems important. In other words, the findings in the studies suggest that although it may, to some extent, be possible to keep your nose in the book, learning while interacting with the computer program may be more valuable.

Text-Picture Processing: Dual Coding and Cognitive Load

The findings of the studies in this book show that there is no difference on cognitive load when comparing a textual and visual manual. This section revisits the use of text-picture combinations in instruction from the view of Dual Coding theory and Cognitive Load theory in light of these findings.

Given the assumptions of Cognitive Load theory (Chandler & Sweller, 1991; Sweller, 1994; Sweller & Chandler, 1994), it was hypothesized that cognitive load would be higher in the visual condition because the screen captures in the manual and on the computer screen present the same information. The redundant character of such instruction would lead to the unnecessary processing of information and thus to an increase of cognitive load.

Given Dual Coding theory (Paivio, 1990; Mayer, 1999), it was expected that working memory was used in a rather optimal way. Dual Coding presupposes that when both textual and visual information is presented, these two types of information are connected in working memory, and this referential connectivity, in turn, contributes to the construction of a strong mental model. This occurs because, according to Mayer and Moreno (1998), working memory consists of two distinct systems: a verbal system, where the textual information is processed, and a non-verbal system that processes visual information.

To conclude, on the basis of the findings that there were no differences on cognitive load, that Dual Coding is the only, or the better explanation for text-picture processing, is rather rash. The remainder of this section presents two critical remarks about Dual Coding theory, and reconsiders the negative impact of redundancy of screen captures assumed by Cognitive Load theory.

Interpretation and application of Dual Coding theory by Mayer and Moreno (1998) and Mayer and Sims (1994) seem rather straightforward. They state as primary assumptions that “(a) working memory includes an auditory and a visual working memory,...(b) each working memory store has a limited capacity,...” (p. 312). It remains unclear, however, what they consider to be the total capacity of working memory. Do, for example, both the non-verbal and verbal system have the same capacity, or do these differ, and if you add the capacity of both, what is the total capacity? It is generally accepted that the storage capacity of working memory is limited to 7 plus or minus 2 elements (Miller, 1956). For working memory according to Dual Coding, this could mean a higher total capacity if both systems have a

capacity of 7 plus or minus 2, which can explain the improved learning and efficiency found in the studies. After all, when using a visual manual, both the verbal and non-verbal system are deployed, leading to potentially more information that can be processed. This assertion is not very plausible. It is more likely that the total of both working-memory systems is limited to 7 plus or minus 2, and that processing of text-picture combinations in the two systems lead to chunks of information. Such a chunk contains several information elements, that for processing in working memory count as only one (Miller, 1956). Departing from the assertion that the capacity of working memory is the same in Dual Coding theory and Cognitive Load theory, the difference between these theories is that Dual Coding theory assumes that simultaneous use of both the verbal and non-verbal memory system causes an advantage over a working memory consisting of only one system. This way, improvements in learning and efficiency can be ascribed to the referential connections between both working memory systems while cognitive load remains similar.

Another issue to be considered with regard to Dual Coding theory is whether both the verbal and non-verbal memory system are indeed used in text-picture processing. Mayer and Moreno (1998) state that textual and visual information are initially both represented in the non-verbal system. The textual information is then translated in auditory form and further processed in the verbal system. That way, both systems are used although both the textual and visual information are of non-verbal nature. When the 'internal' translation of text into auditory (verbal) form does not take place, Dual Coding theory presupposes that both the textual and visual information are processed in the same (non-verbal) memory system. Taking this last view, there would be no difference between the processing of text-picture combinations in Dual Coding theory and Cognitive Load theory, which would explain the similar findings on cognitive load. It does, however, not explain the differences found on learning, training time, and error recovery. The explanation would only hold when the, according to Cognitive Load theory, redundant screen captures would not have negatively impacted text-picture processing.

It can be argued that, although the visual information in the manual and on the computer screen are the same, both are needed to carry out a particular (part of) a learning task. The function verification of screen states serves as good example of this assertion. In case a user checks his or her progress by comparing the screen capture in the manual with the actual screen, the same information is presented (and processed) twice. But without doing so, the operation could never be carried out. The same holds for the other functions. Although the visual information in the manual is equal to that of the computer screen, the screen captures present features that add information, such as improved readability for identifying and locating purposes, and the flow in which the screen is going to change in a series of screen captures for mental model development. The redundant information thus serves important functions. Consequently, it could be argued that because the redundant

screen captures are essential to carry out the task, they do not negatively impact working memory but contribute to learning.

To conclude, it seems that our findings are hard to explain by Cognitive Load theory. The redundancy of screen captures in the visual manual did not lead to an increase on cognitive load when compared to a textual manual. The visual manual even improved learning and error recovery, and brought about a decrease on training time. It is also too early to conclude that Dual Coding theory adequately explains text-picture processing in working memory. Mayer and his colleagues' elaboration on Dual Coding theory leads to some indefiniteness concerning the working memory system in which textual information is being processed, and the capacity of the two memory systems. In the next section, suggestions are given to further study these issues.

Suggestions for future studies

Some reflection is in order to avoid recurring deficiencies in future studies. This section starts with the discussion of two methodological issues that should be taken into consideration when carrying out a new study in the line of research described. This book ends with a presentation of topics for future research that stem from the findings of the studies described and the discussions in this chapter.

Methodological issues

There are two methodological issues that, retrospectively, should be addressed in future studies. They are: the use of a secondary task measure for measuring cognitive load, and the reliability of observations.

In Chapter 6, as an alternative for the self-scoring way of measuring cognitive load, a secondary task measure was used. This secondary task included the shutting off of a sound that was produced by a computer every two minutes. In future studies, using such a secondary task measure needs adjustment on two parts. First, it seems undesirable to use this measure in combination with another one (e.g., self-scoring) that tests the same (e.g., cognitive load). That is, when both measurements in one way or the other interfere with the main task, the level of disturbance may become too high. Moreover, the two measurements may not be mutually exclusive. The awareness of having a certain response time may lead the user into giving a score on the questionnaire that is different than he or she would have given without that knowledge. Second, in future studies, a fixed time interval of presenting the sound should not be used to prevent expectancy of the distraction, or to prevent automated responses to occur.

A second methodological issue concerns the handling of the reliability of observations. Measurements in research should be objective and reliable. The most objective measures used in the experiments in this book are (log-)registrations to measure time, cognitive load, and motivation. Based on, for example, registered

mouse clicks or a filled in nine-point scale, an exact and uni-interpretible measure was obtained. Tests that measure problem solving or identification ability are somewhat less objective because they require some sort of judgement. Based on strict directives for scoring the tests, a desired objectivity level is guaranteed. Gathering data by observations becomes somewhat more difficult in terms of objectivity. In the type of research presented in this book, observation sometimes seems the only available method to gather the desired data. Especially in Chapter 4, the difficulty of observing resulted in just acceptable interrater reliability. This causes some concern. The speed of the behavior to be observed, in this case looking up from the manual to the screen, was quite high, and the movements were rather subtle. That is, it sometimes appeared to be difficult to catch up with the rate of action steps users performed, and thus to observe the exact amount of switches they made. In turn, because the head-movements were not directed exclusively to manual and screen, but also to manual and keyboard, and to keyboard and screen, reliability of this measure was negatively influenced. To cope with the difficulties of tempo of task execution and subtlety of head-movements, future studies should video-tape the user as a backup for observations that are missed or considered doubtful.

Research topics

Finally, guided by the main organization of this book, suggestions for future research topics will be given from the perspectives of instructional design (the four functions of screen captures) and text-picture processing.

The setup of a next study on the four functions of screen captures in software manuals seems obvious. After the discrete examination of the functions, it should be tested if the merits of each function remain intact or even become reinforced when integrating support by screen captures for the four functions in one manual. The setup of such an experiment would be a synthesis of the experiments in this book. A textual and visual manual would be compared in a rather extensive experimental training session where (1) switching attention would be measured through (video-taped) observation, (2) the development of a mental model and (3) identification of window elements and objects would be measured by post-tests, and localization by measuring training time through log-registration, and (4) verification of screen states would be measured by verification time, and verification quality by error-recovery of intentional errors by log-registration and/or observation.

From the text-picture processing perspective, one topic seems especially worth examining in future research. That is, to study whether textual information is mainly processed in the verbal or the non-verbal working memory system. For this, instruction in six forms should be designed. Three of these six are uni-modal instructions: visual instruction, auditory instruction, and textual instruction. According to Dual Coding theory, learning effects of these three instructions are expected to be equal. The visual instruction will be processed in the non-verbal

mode, the auditory instruction in the verbal mode, and the textual instruction in one of the two modes (which one remains unknown so far).

Three multi-modal instructions are designed based on combinations of the three uni-modal instructions: visual-auditory instruction, visual-textual instruction, and auditory-textual instruction. It is expected that the visual-auditory instruction will produce better results than either one of the uni-modal instructions. After all, the visual-auditory instruction is processed in both the verbal (auditory) and non-verbal (visual) memory system and the referential connections between both systems lead to enhanced learning. The learning effects for the visual-textual and the auditory-textual instruction will show in which memory system the textual information is being processed. It is expected that one of these two instructional formats will produce learning effects that are equal to the multi-modal (visual-auditory) instruction and the other equal to the uni-modal (visual, or auditory, or textual) instruction. Learning effects on uni-modal level show that the instruction is processed in one of the two systems, whereas effects on multi-modal level show dual processing. More specifically, if the visual-textual instruction produces multi-modal learning effects, textual information has been processed in the verbal system, if the auditory-textual instruction leads to multi-modal learning effects, textual information has been processed in the non-verbal memory system.

Finally, in this study, learning effects for only one of the three uni-modal instructions need to be obtained to find out in which memory system textual information is processed. This experiment therefore needs a minimum of four conditions (one uni-modal and the three multi-modal instructions). Because Dual Coding theory does not give explanations or predictions for differences in learning effects of various forms of uni-modal instruction (e.g., visual, auditory, or textual), it is however useful to obtain the learning effects of all three uni-modal instructions, to assess whether these instructions indeed lead to the same learning.

References

- Chandler, P., & Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognition and Instruction*, 8, 293-332.
- Mayer, R. E. (1999). Designing instruction for constructivist learning. In: Reigeluth, C.M., *Instructional-Design theories and models Volume II: A new paradigm of instructional theory*. Mahwah, New Jersey: Lawrence Erlbaum Associates.
- Mayer, R. E., & Moreno, R. (1998). A split-attention effect in multimedia learning: Evidence for Dual processing systems in working memory. *Journal of Educational Psychology*, 90, 213-330.
- Mayer, R. E., & Sims, V. K. (1994). For whom is a picture worth a thousand words? Extensions of a Dual-Coding theory of multimedia learning. *Journal of Educational Psychology*, 86, 389-401.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63, 81-97.

Paivio, A. (1990). *Mental Representations: A Dual Coding Approach*. New York: Oxford University Press.

Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and Instruction*, 4, 295-312.

Sweller, J., & Chandler, P. (1994). Why some material is difficult to learn. *Cognition and Instruction*, 12, 185-233.

Nederlandse Samenvatting (Dutch Summary)

Dit proefschrift gaat over het gebruik van visualisaties in instructie. Visualisaties en instructie zijn er in allerlei typen. Het type instructie dat hier wordt onderzocht is een handleiding om te leren werken met een computer programma. Het type visualisaties dat in zulke handleidingen het meest wordt gebruikt zijn afbeeldingen van het beeldscherm. Het specifieke onderwerp van dit proefschrift is dus het gebruik van schermafbeeldingen in computer handleidingen.

Het onderzoek naar het gebruik van schermafbeeldingen in computer handleidingen richt zich op het beantwoorden van twee vragen. De eerste vraag is hoe schermafbeeldingen moeten worden ingezet om ervoor te zorgen dat een gebruiker snel en veel van een handleiding leert. Op basis van het beantwoorden van deze vraag kunnen aanwijzingen worden gegeven over hoe schermafbeeldingen gebruikt kunnen worden om handleidingen te verbeteren. De tweede vraag is hoe een combinatie van visuele en tekstuele informatie in het geheugen wordt verwerkt. Twee beschikbare theorieën over dit verwerkingsproces lijken in tegenspraak. De Dual Coding theorie voorspelt dat gebruikers meer zullen leren van een handleiding waarin tekst en schermafbeeldingen tegelijk worden aangeboden, terwijl de Cognitive Load theorie stelt dat leren door zo'n combinatie wordt belemmerd.

Schermafbeeldingen om handleidingen te verbeteren

Het doel van een computer handleiding is dat gebruikers leren werken met een computer programma in een zo kort mogelijke tijd. Een belangrijke aanleiding om te onderzoeken of schermafbeeldingen een handleiding kunnen verbeteren was de observatie dat handleidingen in het algemeen steeds visueler werden. Vooral in succesvolle, goed verkopende handleidingen lijken schermafbeeldingen inmiddels net zo gebruikelijk als tekst. Of het gebruik van schermafbeeldingen er ook voor zorgt dat gebruikers er meer of sneller door leren was onbekend. In het eerste experiment is dit onderzocht (zie hoofdstuk 2). In dit experiment zijn twee visuele handleidingen, afgeleid van commercieel succesvolle handleidingen, vergeleken met een tekstuele handleiding. Het verschil tussen de twee visuele handleidingen was dat in de ene alleen afbeeldingen van het gehele beeldscherm werden gebruikt en in de andere voornamelijk afbeeldingen van gedeelten van het beeldscherm. Het bleek dat gebruikers van de handleiding met de gedeeltelijke schermafbeeldingen minder leerden dan gebruikers van zowel de tekstuele handleiding als de handleiding met gehele schermafbeeldingen. Tussen deze laatste twee bestond geen verschil. Hieruit kan worden geconcludeerd dat het blijkbaar van belang is welk type schermafbeelding gebruikt wordt en dat het gebruik van schermafbeeldingen niet noodzakelijk leidt tot een betere handleiding. Immers, de tekstuele en visuele handleiding met gehele schermafbeeldingen werkten even goed.

De uitkomsten van dit eerste, exploratieve experiment gaven aanleiding om het gebruik van schermafbeeldingen in handleidingen op een wat andere wijze te gaan beschouwen. In de onderzochte visuele handleidingen waren de schermafbeeldingen ingezet volgens een genre-benadering. Hiermee wordt bedoeld dat de schermafbeeldingen consistent werden aangeboden ongeacht het doel. Juist het inzetten van schermafbeeldingen voor het ondersteunen van één of meer specifieke doelen geeft goede mogelijkheid een handleiding te verbeteren. Een gedetailleerde analyse van de instructie in een handleiding volgens deze doelgerichte benadering resulteerde in een raamwerk waarin vier specifieke functies voor het gebruik van schermafbeeldingen in handleidingen worden onderscheiden (zie hoofdstuk 3). Schermafbeeldingen ondersteunen:

- het wisselen van de aandacht tussen de handleiding en het beeldscherm
- de ontwikkeling van een mentaal model van het computer programma
- de identificatie en lokalisatie van objecten op het beeldscherm
- het verifiëren van informatie op het beeldscherm

In drie experimenten is onderzocht of het gebruik van schermafbeeldingen voor het ondersteunen van deze functies inderdaad leidde tot een verbetering van de handleiding. In deze experimenten werd telkens een tekstuele handleiding vergeleken met een visuele handleiding waarin schermafbeeldingen werden gebruikt voor het ondersteunen van een specifieke functie. In elk experiment werd aan de gebruikers gevraagd enige tijd met een handleiding te werken. Tijdens deze oefensessies werd hun gedrag vastgelegd door observatie of log-registratie en werd de tijd gemeten die men nodig had om (delen van) de handleiding door te werken. Na de oefensessie werd een aantal tests afgenomen.

Uit het experiment waarin geobserveerd werd hoe vaak gebruikers opkijken van de handleiding naar het beeldscherm, bleek dat gebruikers van zowel een tekstuele als een visuele handleiding zeer frequent hun aandacht wisselden (zie hoofdstuk 4). Op vrijwel alle momenten waarop verwacht kon worden dat de aandacht werd gewisseld gebeurde dat ook. Dit betekende dat de handleiding met betrekking tot deze functie nauwelijks was te verbeteren.

Het gebruik van schermafbeeldingen ter ondersteuning van de ontwikkeling van een mentaal model had een duidelijk effect. Gebruikers van een visuele handleiding ontwikkelden een sterker mentaal model dan gebruikers van een tekstuele handleiding (zie hoofdstuk 5). Dit mentale model kwam tot uitdrukking in scores op tests waarin werd gemeten of gebruikers aan de hand van hun kennis over het computer programma problemen konden oplossen.

Hetzelfde gold voor de identificatie en lokalisatie van objecten op het beeldscherm. Ook hier bleek een visuele handleiding beter te werken dan een tekstuele handleiding (zie hoofdstuk 5). Gebruikers van de visuele handleiding scoorden beter op tests die hun kennis over de objecten op het beeldscherm maten (identificatie) en waren sneller in het vinden van objecten op het beeldscherm (lokalisatie).

Gebruikers van een visuele handleiding bleken ook beter in het verifiëren van informatie dan gebruikers van een tekstuele handleiding (zie hoofdstuk 6). Op basis

van de informatie in de handleiding moesten gebruikers controleren of deze gelijk was met de informatie op het beeldscherm, en zo nodig aanpassen. Gebruikers van een visuele handleiding bleken niet alleen minder fouten te maken bij deze verificatie-taken, maar deden dit ook in minder tijd dan gebruikers van een tekstuele handleiding.

Verwerking van tekst en afbeeldingen in het geheugen

De Dual Coding theorie stelt dat het werkgeheugen bestaat uit twee delen; een deel waarin verbale informatie wordt verwerkt en een deel waarin non-verbale informatie wordt verwerkt. Tekst wordt volgens deze theorie verwerkt in het verbale geheugendeel en afbeeldingen in het non-verbale geheugendeel. Beide geheugendelen staan met elkaar in verbinding. Wanneer beide geheugendelen tegelijkertijd benut worden en vervolgens aan elkaar refereren, ontstaan sterke verbindingen die resulteren in een leereffect. Dit leereffect wordt verwacht sterker te zijn wanneer beide geheugendelen worden gebruikt dan gebruik van elk geheugendeel afzonderlijk. Voor het leren met handleidingen betekent dit dat een visuele handleiding, waar tekst en afbeeldingen tegelijkertijd worden aangeboden en aan elkaar gerelateerd zijn, zal leiden tot een sterker leereffect dan een tekstuele handleiding.

De Cognitive Load theorie stelt dat het werkgeheugen bestaat uit slechts één deel. De beperkte capaciteit van dat geheugen moet zo verstandig mogelijk worden benut. Dit betekent onder andere dat informatie maar één keer moet worden aangeboden. Het meerdere keren aanbieden van dezelfde informatie leidt tot gebruik van geheugenruimte die beter door andere, of nieuwe informatie gebruikt had kunnen worden. Voor het leren met handleidingen betekent dit dat een visuele handleiding het leren zal belemmeren. De schermafbeeldingen in de handleiding zijn gelijk aan de informatie op het beeldscherm. Er wordt geheugenruimte gebruikt om twee keer dezelfde informatie te verwerken.

In twee experimenten is onderzocht of de geheugenbelasting tussen gebruikers van de tekstuele en visuele handleiding verschilde (zie hoofdstuk 5 en hoofdstuk 6). De geheugenbelasting werd gemeten door gebruikers te vragen hoe moeilijk ze bepaalde taken vonden. Deze score werd weergegeven op een glijdende schaal van 'zeer moeilijk' naar 'zeer makkelijk'. In één experiment (zie hoofdstuk 6) werd tevens de reaktietijd gemeten. Dit werd gedaan door gebruikers een geluid te laten uitzetten dat zo nu en dan werd aangeboden terwijl ze aan het werken waren met de handleiding. Een lange reaktietijd werd gezien als een hoge geheugenbelasting en een korte reaktietijd als een lage geheugenbelasting.

Op alle metingen bleek de geheugenbelasting van de gebruikers van een visuele handleiding gelijk aan die van de gebruikers van een tekstuele handleiding. Deze uitkomsten, in combinatie met de gevonden leereffecten, ondersteunen in eerste instantie de Dual Coding theorie. Er wordt door gebruikers van de visuele handleiding immers meer geleerd bij gelijkblijvende geheugenbelasting.

Beantwoording van de onderzoeksvragen

De eerste onderzoeksvraag was hoe schermafbeeldingen moeten worden ingezet om ervoor te zorgen dat een gebruiker snel en veel van een handleiding leert. De uitkomsten van het onderzoek in dit proefschrift tonen aan dat een doelgerichte benadering bij het inzetten van schermafbeeldingen leidt tot een verbetering van de effectiviteit en efficiency van de handleiding. Door met schermafbeeldingen specifieke functies te ondersteunen wordt meer geleerd in minder tijd.

De tweede onderzoeksvraag was hoe een combinatie van visuele en tekstuele informatie in het geheugen wordt verwerkt. Op basis van de onderzoeksresultaten ligt een verklaring volgens de Dual Coding theorie voor de hand. Deze conclusie kan echter alleen worden getrokken met het nodige voorbehoud. De Dual Coding theorie gaat er van uit dat tekstuele informatie in het verbale geheugendeel wordt verwerkt terwijl tekst op zich non-verbaal is. De aanname is dat de non-verbale tekst als het ware wordt uitgesproken en vervolgens in het verbale geheugendeel wordt verwerkt. Deze aanname dient nader te worden onderzocht. Indien tekst en afbeeldingen toch in hetzelfde (non-verbale) geheugendeel blijken te worden verwerkt, dan houdt de verklaring volgens de Dual Coding theorie geen stand.

Appendix A: Text Example Page for Chapter 2

The **SERVIVE** project

SIMQUEST 1.0

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**

Appendix B: Visual-Part Example Page for Chapter 2

The **SERVIVE** project

SIMQUEST 1.0

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.

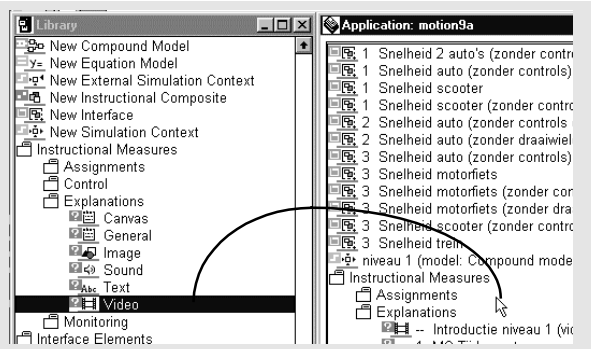
Dragging and dropping an explanation

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*



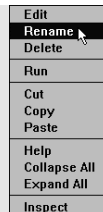
Naming an explanation

To keep it clear what your elements contain, you can give the explanation a meaningful 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**



Appendix C: Visual-Full Example Page for Chapter 2

The **SERVIE** project

SIMQUEST 1.0

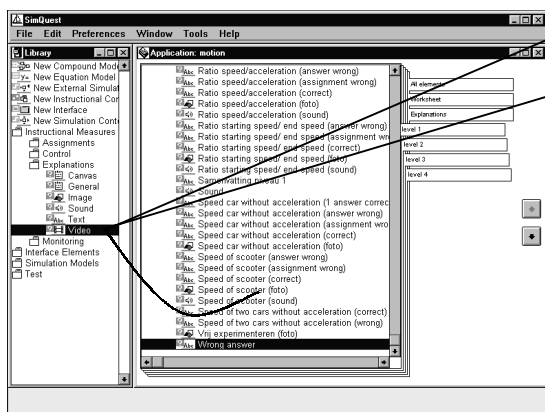
Adding explanations to the application

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

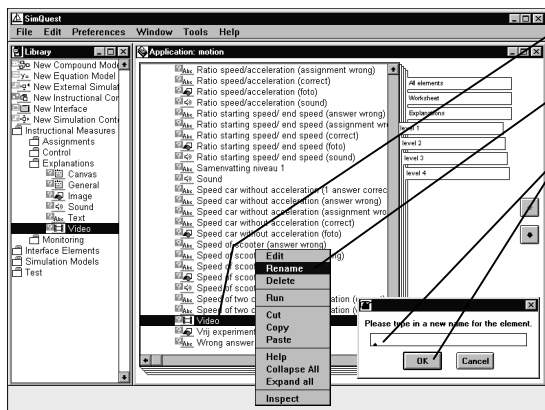
You already decided which explanation element you are going to use, you can take it from the library and put it into your application.



- 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 meaningful 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**

Appendix D: Design dimensions

Coverage refers to the desktop, window, window element, or object displayed in the screen capture. One extreme position on this dimension is a screen capture that shows the full screen. The other extreme position is a screen capture that presents a single object (icon, button, symbol, menu option). Examples of such objects are window controls, such as the maximize and minimize button, and menu items. Between these two extremes fall screen captures such as: a set of cascading windows, a single window, a title bar, a menu, and a dialog box.

Positioning refers to the placement of text and screen captures in relation to one another, reflecting their relationship or interdependence. Screen captures can be presented in such a way that they are visibly separated from the text, or they may be integrated within the text. A separated display requires a lay-out with two or more columns. A typical example is shown in Figure 1. An example of an integrated display is shown in Figure 2.

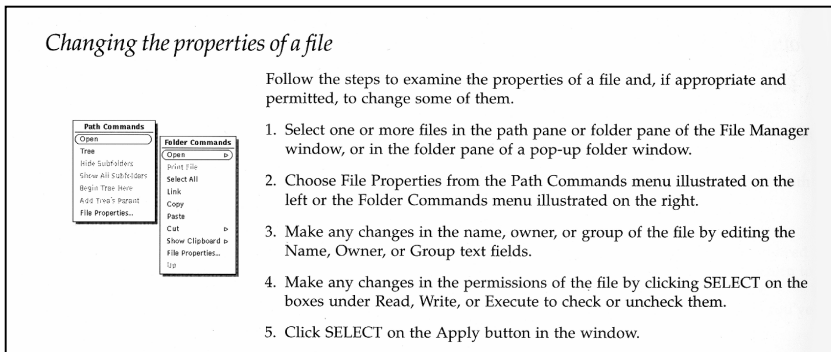


Figure 1. A separated display of screen captures and corresponding text

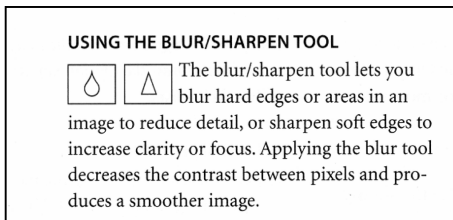


Figure 2. An integrated display of screen captures and corresponding text

Size refers to the reduction rate of the desktop, window, or object in the screen capture as compared to its actual screen size. The choice for a certain reduction rate depends on considerations such as the consumption of space, legibility, company standards, and relationships between screen captures.

Cueing refers to the presence or absence of signaling techniques that draw the users' attention to relevant window elements or objects. Signaling techniques include the use of (colored) hairlines, circles, callouts, and blurring. A descriptive caption or key word may be presented along with the cue. Cueing should never confuse the user as to whether the cue should be part of what is shown on the actual screen. Figure 3 displays an error-prone example.

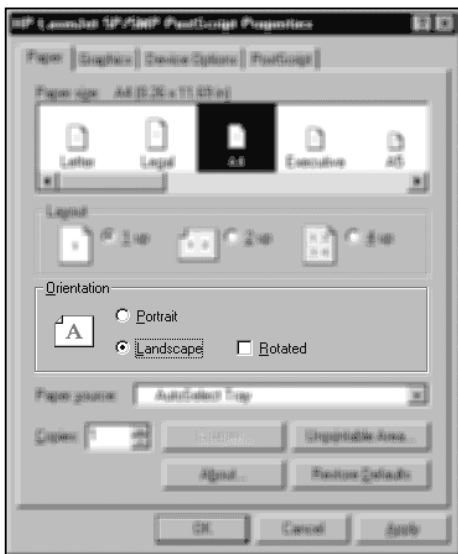


Figure 3. Cueing should never lead to a screen capture the user might mistake for the real window

Figure References

1. Sun Microsystems. (1991). *Desktop SPARC: Sun System User's Guide*. Mountain View, CA: Sun Microsystems Inc., pp. 86.
2. Adobe Systems. (1994). *Adobe Photoshop 3.0 User Guide*. Edinburgh, United Kingdom: Adobe Systems Europe Limited, pp. 107.