Dumb Screen Dumps, Smart Screen Captures – A Case Study of Screen Captures in Software Documentation

Jouni Merikari
University of Tampere
The School of Modern Languages and Translation Studies
Translation Studies (English)
Pro Gradu Thesis
January 2008

Tampereen yliopisto
Käännöstiede (englanti)
Kieli- ja käännöstieteiden laitos

MERIKARI, JOUNI: Dumb Screen Dumps, Smart Screen Captures – A Case Study of Screen Captures in Software Documentation

Pro gradu -tutkielma, 67 sivua, 3 liitesivua, suomenkielinen lyhennelmä 7 sivua
Tammikuu 2008

Tutkimuksen tavoitteena oli selvittää miten ohjelmistotuotteiden dokumentaatiossa käytetään tietokoneohjelmista otettuja kuvaruutukaappauksia ja kuinka hyvin käytetyt kuvaruutukaappaukset tukevat käyttäjien tarpeita.

Yleisenä viitekehyksenä kuvallisen informaation suunnittelulle esiteltiin Carlinerin kolmiosainen informaationsuunnittelumalli, joka jakautuu fyysiseen, kognitiiviseen ja affektiiviseen alueeseen. Kuvaruutukaappausten rooleja ohjelmistodokumentaatiossa oli valittu kuvaamaan van der Meij'n ja Gellevij'n teoria kuvaruutukaappausten rooleista ja näihin rooleihin vaikuttavista suunnittelun osa-alueista.

Aineistoksi valittiin ohjelmistotuotteiden loppukäyttäjille tarkoitettuja käyttöohjeita, jotka olivat vapaasti saatavilla Internetistä. Kriteerinä oli myös se että kuvatut ohjelmistotuotteet eivät olisi pitkälle erikoistuneita ammattiohjelmia. Tutkimuksessa otettiin tarkasteltavaksi kymmenen kuvaruutukaappauksen satunnaisotanta jokaisesta dokumentista. Näin oli mahdollista analysoida myös pitkiä dokumentteja, joiden sisältämien kymmenien tai satojen kuvaruutukaappausten analysoiminen kokonaan olisi ollut muuten liian työlästä.

Metodi tutkimukseen saatiin muokattua van der Meij'n ja Gellevij'n teoriassa esitellyistä suunnittelun osa-alueista. Aineiston kuvat analysoitiin suhteessa jokaiseen osa-alueeseen ja tulokset esitettiin kvantitatiivisessa muodossa, sekä kokonaisuutena että dokumenteittain. Lopuksi tulokset koottiin yhteen ja esitettiin osana Carlinerin informaatiosuunnittelumallia.

Tutkimus osoitti, että vaikka dokumenttien välillä oli selkeitä eroja johtuen ko. dokumenteissa valitusta tavasta esittää informaatiota, niin oli mahdollista havaita mitkä alueet parhaiten tai heikoiten tukivat käyttäjien tarpeita. Analyysin lopputuloksena voidaan sanoa, että heikoiten käyttäjää tuettiin niillä kuvaruutukaappausten suunnittelun osa-alueilla, jotka kuuluvat ns. affektiiviseen suunnitteluun. Seuraavaksi eniten ongelmatapauksia oli fyysisen suunnittelun puolella, mutta kognitiivinen suunnittelu otti käyttäjät huomioon suhteellisen hyvin.

Tutkimustuloksista oli johdettavissa johtopäätöksiä siitä mihin informaationsuunnittelun osa-alueisiin pitäisi kiinnittää lisää huomiota käytännön kirjoitustyössä ja mitä yksittäisiä ongelmia tekninen viestijä voi kohdata. Löydösten pohjalta havaittiin myös potentiaalisia jatkotutkimuksen kohteita, joista tärkeimmät olivat tekstin ja kuvan suhteen tutkiminen, kulttuurin vaikutus ja erilaisten suunnittelustrategioiden testaaminen käyttäjillä.

Avainsanat: tekninen viestintä, kuvat, ohjelmat, ohjelmistot, ohjelmistodokumentaatio

TABLE OF CONTENTS

TABLE OF FIGURES

1 Introduction

Screen captures are snapshot images of computer programs that are used in software documentation to guide the user. They should be studied for two main reasons: screen captures are very common in technical communication and they have specific roles that we do not fully understand.

The relationship between text and picture has been studied, for example, in usability studies and translation studies. Pictures have also been studied in technical communication, and technical communication acknowledges the multitude of parallel and intersecting fields of study, for example, cognitive psychology, linguistics, and human-computer interaction (HCI). The human-centred aspect of technical communication is inherently important and this is why I have chosen to study software documentation in the light of theories that emphasise user cognition and motivation.

We who work in the field of technical communication receive little support for designing visual and textual information for users. I have worked as a technical writer for over seven years. During that time I have been offered one training that was remotely related to graphics; the training handled mainly graphics tool issues and customer requirements. I know the situation is the same with colleagues in other countries: Germany, Malaysia, India, China, and the United States. Technical communicators are given training and instructions on how to conform to customer requirements and expectations, when we should be exceeding the customers' expectations. This is not to say that presenting visual information to users has not been studied. There are proponents for a more user-cantered approach to visualising information, e.g. Brasseur, but he concentrates more on the critique of the established technical visual genres and the cultural and social constraints that surround the area of designing visual information. He fully acknowledges that a cultural study such as his is meant to question, not to give final answers:

> I adopt this method knowing full well that answers are the usual end of a cultural investigation. However, such answers are not likely to be found in a critique such as mine because I focus on the circumstances surrounding the genre. (2)

Brasseur also mentions Tufte as one of the persons who has particularly influenced his work (ix). Tufte is an expert on presenting information-rich graphics, for example charts and diagrams, but his audience is not specifically technical communicators, and his works do not address the daily problems that we face in our profession. There are textbooks on technical communication that mention graphics (see for example McMurrey, or Kolstelnick & Roberts, or Price), but these usually contain little information on designing visual information. Some focus more on the ethical choices that technical communicators face. Rossner mentions that students (of technical communication) are given instructions, for example, on how to place figures and label them descriptively, but we fail to make them aware of the choices that "shape the pictures they make and see" (392). At the same time, visual literacy is becoming more and more important as we gradually move from the dominance of writing and the book, towards the dominance of the image and the screen (Kress 1).

Why this shift in dominance? I believe the rise of personal computing and especially graphical user interfaces (GUIs) in the last 20-30 years is the main reason, though television has probably had a significant impact in its 100 years history. Homby gives an interesting view into to the history of the GUI. The first GUI, used in Alto computer at Xerox PARC, was designed already in 1973. Xerox gave Apple Computer engineers access to the Alto computer in exchange for stocks. Later many of the Xerox engineers went to work for Apple because they were frustrated with the Xerox executives who did not want to release the new technology. The real breakthroughs in personal computing came as Apple and several other companies took advantage of the Alto design in the mid-1980s. Since then GUIs have become more and more common. They surround us everywhere from homes to workplaces and hospitals to supermarkets. We are presented with a challenge to design visual information in a new age where the image and visual literacy are predominant. Kress describes the effects that this shift may have:

> The combined effects on writing of the dominance of the mode of image and of medium of the screen will produce deep changes in the forms and functions of writing. This in turn will have profound effects on human, cognitive/affective, cultural and bodily engagement with the world, and on the forms and shapes of knowledge. (1)

Here Kress brings up two themes that are important in this thesis: cognitive and affective effects. Both of these are part of the three-part framework for information design presented by Carliner.

His framework takes into account three important factors of information design: physical, cognitive, and affective (564). Carliner's model is a high-level framework, a comprehensive overview, which can support a more detailed theory describing screen captures.

A recent theory presented by van der Meij and Gellevij offers a theoretical background for understanding screen captures on a more detailed level. Their theory introduces a number of analyses conducted on actual users. It can also be used to explain how screen captures work in Carliner's three-part framework of design: helping users find and understand information, and motivating users. The theory also presents a taxonomy of the main roles (ways in which screen captures help user cognition) and design areas (respective physical properties like coverage or size) of screen captures (van der Meij and Gellevij, b). While this theory is helpful, understanding how visual information and screen captures help users to find, understand, and use information is not an easy task. Screen captures have to be studied in detail to understand what their role is in software documentation. The theory, though originally based on their preliminary findings, gives us an insight into how users could be affected and motivated by screen captures. Recently, they have further extended and bolstered their theory (van der Meij and Gellevij, a). Together these studies give us sufficient theoretical background to analyse software documents to see how well they support users' needs.

Although the emphasis in this thesis is on screen captures, I will comment the relationship between verbal and visual information to some extent. For example, the positioning of pictures can be an important factor (van der Meij, a: 1). The relationship of text and picture is not a simple subject to study, and many different factors affect the user's cognitive process. It is debatable which information is more important: verbal or non-verbal? The dual coding theory proposed by Paivio separates verbal and non-verbal processing, but gives them equal weight:

> Human cognition is unique in that it has become specialized for dealing simultaneously with language and with nonverbal objects and events. Moreover, the language system is peculiar in that it deals directly with linguistic input and output (in the form of speech or writing) while at the same time serving a symbolic function with respect to nonverbal objects, events, and behaviors. Any representational theory must accommodate this dual functionality. (53)

Dual coding theory has been applied to many different cognitive processes, such as language, learning, and problem-solving. Van der Meij and Gellevij propose that because of dual coding, users of software documentation can process more information if they are given information through both of these channels, and they also propose that users can learn better when words and pictures share important features (a: 2-3).

1.1 Aim

The aim of this study is to use the theory of the screen capture roles to analyse how screen captures are used in software documents today, and how well they take into account different aspects of information design.

This is the rationale of the study:

1. To identify the roles and design areas of screen captures in software documentation, and the aspects of information design that are important to users. This will be done by examining and expanding on theories presented by Carliner, and van der Meij and Gellevij. This part forms the theoretical basis for the analysis.

2. To evaluate how well screen captures used in software documentation support the roles of screen captures. This part forms the analysis part.

This idea of evaluating is also present in van der Meij's presentation in the 47th conference of the Society for Technical Communication:

> The taxonomy [of screen captures' roles and design areas] can facilitate design discussion and evaluations. It can form the basis for an articulated discussion on when and how to present screen captures in software documentation. The taxonomy can also be used to test predictions about the effects of screen captures on the user's thoughts and actions. (a: 1)

The design areas of screen captures presented in the theoretical part of the study will be used as a starting point for the evaluation, as van der Meij proposes. In practice this means that all the design areas of a set of screen captures will be analysed, and the results will then be presented in the larger framework of information design – that of Carliner's. The focus is on the three aspects of information design, but I do not study usability or users, only the screen captures. The focus is also strictly on the content, not in ways to produce the content. Carliner mentions in his article on information design:

> This is an exciting time for technical communicators. We're moving from a focus on the tools used to produce content, like help authoring tools and desktop publishing programs, to a focus on the content itself. (561)

He too senses the problem of presenting tool-specific instructions to technical communicators. Even if we are given very accurate instructions and have the correct tools, a poorly designed product still leads to poor documentation.

Problems arising from technical solutions, size of the documents (large images, etc.) or tools used to present and produce technical documents are not covered in this thesis. Also, the medium (paper, screen, or online) where the documentation is published is not studied because this would introduce many medium-specific or tool-specific issues that would have to be discussed. This is an important omission because otherwise each medium should be addressed separately; a subject better suited for a more extensive study of figures in technical communication. The screen captures found in the material will be analysed as they are, in relation to the document (be it paper or screen), and medium-specific issues will only be pointed out if they cause problems in analysing the material.

There are other omissions, too. Although this thesis fits into the larger structure presented by Carliner, it should be emphasised early on that this thesis is concerned with the **main** roles and **main** design areas, a point that is also clearly stated by van der Meij and Gellevij in their theory. I admit that there can be other less significant roles and design areas, but here the focus is on the macro level.

The results of this study could perhaps be used to improve the quality of software documentation, and even give practical advice. If it is found that different problem areas can be identified using van der Meij and Gellevij's theory, then the discussion part of this thesis will aim to present probable solutions to the problems in current software documents. Like this thesis, the solutions suggested should be independent from media or any technical tools used to produce or edit screen captures. Solutions should be guidelines on how to make screen captures more effective in technical communication. Possible problem groups found could form a typology, for example, for analysing software documents.

1.2 Organisation of This Study

This study is divided into six main chapters:

Chapter 1 is this introduction.

Chapter 2 will discuss the general nature of screen captures in software documentation: how screen captures have been defined before and what implications the digital nature of screen captures has.

Chapter 3 introduces the theoretical background of this thesis in detail. I will first introduce information design framework presented by Carliner, and proceed to explain the theory of screen capture roles and design areas, as presented by van der Meij and Gellevij.

Chapter 4 introduces the material and the methods used in the analysis. I will describe the criteria for choosing five case documents and explain how individual screen captures are chosen. The description of the method will show how the theory of screen capture roles will be applied when analysing screen captures.

Chapter 5 presents the results of the analysis and establishes how the results of this analysis fit into the three-part framework of Carliner.

Chapter 6 is the final chapter in the thesis. It will be reserved for the conclusion and discussion of the results. Also, potential areas for further study will be identified.

This thesis also contains a glossary and an appendix. The glossary contains a list of the terminology used and definitions, and the appendix shows a list of all the findings of the analysis phase.

2 Screen Captures and Software Documentation

Before I can explain how screen captures are used and how they can affect users, I need to define what I mean with software documentation. The term software documentation applies to a whole range of different kinds of documents. 'Documentation' can be used in technical communication for printed and electronic manuals and other documents delivered to the customer, but it can also refer to technical descriptions not meant for the users, project management documents, or even marketing brochures (Lahti 7-8). All these different types have slightly different audiences, contents, purposes, and authors, as can be seen in the table below:

Table 1 Types of software documentation (Lahti 8)

Type	Audience	Content	Purpose	Producer (writer)
Design Documents Also: technical documentation, technical specifications, product documentation	Software developers, technical writers, technical support.	Detailed descriptions of how the software code was designed and implemented.	To enable modularity with other software. To enable continued development even if the original developers leave.	Software developers.
User Documentation Also: Customer documentation, end user documentation, product documentation.	End users, help desk, technical support, customers and purchasers, new software developers, marketing staff	Descriptions about the software and its features; instructions for using the software; information about compatibility issues.	To make the software more professional and fulfill consumer law requirements. To inform users about tasks with the software To reduce the number of technical support requests. To make users more satisfied with the product.	Technical writers; sometimes also software developers, training people and marketing communicators.
Product documentation	Project's steering group, project members, product managers, marketing officers	Descriptions of the target market and target audience for the product; comparison of the development effort with results.	To help assess if a project is worth executing. To convey information about the project to various interest groups.	Project manager and project members.
Marketing material Also product documentation, brochures, fact sheets.	Customers, purchasers, sometimes also end users.	Descriptions and comparisons of software capabilities; examples of how the product saves users' time and money.	To help convince customers to buy the product.	Marketing communicators, technical writers.

In this thesis, I will concentrate on software documentation. The term 'software documentation' is used here to refer to user documentation only. Other types of documents are beyond the scope of this thesis, although many findings could probably be applied to them if they use screen captures in instructions.

The design of these communication products, user documents, is often called technical communication, document design, or information design[1]. I prefer the term 'information design' because it broadens the role of technical communication beyond traditional boundaries of writing, and it emphasises technical communicator's role as a professional of design. In my professional opinion, the role of the technical communicator is not only to document, but also to actively work alongside other specialists to make more usable products: to guide and motivate the users. It is more than technical writing. It is customising information for a specific user group (or groups) who are using a specific product.

Software documents are utility texts. I have adopted this term from Pilto and Rapakko (37-39) who give 'utility text' as one possible name for documents that facilitate or enable the use of another product or service. Another name could be 'necessary texts' that is used by Cook who defines them as purpose-oriented messages written by specialists, often constrained by genre conventions, and having practical and observable outcome (15). Software documents do not have a purpose by themselves; they are used so that the user can use or operate another product. This product is a computer program (software), as opposed to computer equipment (hardware).

2.1 Screen Captures

Screen captures are screen snapshots of computer programs that are used in software documentation to guide the users. In this chapter, I will expand on this basic idea and present other, mainly complementary, views.

Many of the definitions for screen captures are still from the early days of computing. In the Oxford Dictionary of Computing, published in 1996, there is no entry for the term 'screen capture'. Instead, the dictionary includes a somewhat older term: 'screen dump'. The entry reads:

[1] For a more detailed discussion, see e.g. Schriver (4-11).

> Screen dump.
>
> A way of transferring the entire graphical or textual contents of a display screen to a printer. Each *pixel is of the display appears as a dot of suitable density on the printer. Color screens can be dumped to color printers.

This definition sounds old because it does not take into account that the screen dump can be transferred to other media as well, not just paper. More recent definitions are closer to the mark. For example, Horton uses the term 'snapshot' and presents the following definition:

> Screen snapshots are literal representations of what appears on the user's screen or in a window of that screen. (146)

This definition is not necessarily limited to a specific medium and it states that a screen capture is a representation. Both of the above definitions are combined in a more recent definition given in the American National Standard for Telecommunications – Telecom Glossary 2000:

> In computers, the process or act by which the data currently displayed on a monitor, usually representing a single frame of information, are stored or processed in a graphical format.
>
> Note: A screen capture thus represents an instantaneous "snapshot" of the state of the display. [underlining mine]

This definition may seem a bit technical and complicated, but it is more accurate than the ones presented by the Oxford Dictionary of Computing and Horton. Here the definition better conveys the idea of screen capture as digital information. Screen capture itself is not a tangible object; it is information that can be manipulated (stored, duplicated, modified, searched, etc.) in the same way as all digital information.

It must be remembered that understanding what screen captures are and knowing what they do does not necessarily give us power over them. There always seems to be a certain factor of uncertainty to images. Images, like technologies, are made by people, but at the same time are thought to be "out of control" (Mitchell 6). Mitchell means that it is difficult to predict how a picture will be perceived or understood by different users. It is a situation very similar to cognitive processes associated with the use of a language, with at least one major exception: pictures leave even more room for interpretation. Humans share cognitive processes but there is cultural variation and variation between different users. Predicting these cognitive processes and their outcomes is vital in our profession. Ganier, for example, proposes that technical communicators can greatly improve the design of procedural documents by taking into account various cognitive and affective design factors (15).

What then needs to be taken into account when studying how screen captures affect users? In software documentation screen captures have a very specific function, but one can never wholly predict how these "frames of information" interact with human needs and expectations. What we can predict is how screen captures affect human cognition in general. There have been many empirical studies on users of screen captures and software manuals. For example, the effects of screen captures on user cognition have been tested on novice users and it has been found out that screen captures alleviate problems with memory load and visual scanning (van der Meij and Gellevij, a).

From previous studies, including the one mentioned above, it is evident that screen captures have definite and measurable effects on user cognition, but how do they actually achieve these effects? It has been suggested that screen captures have a set of main roles and design areas that affect the user's ability to understand instructions and the structure of a software product (van der Meij and Gellevij: b). A software document that uses only verbal instructions does not affect user cognition in the same way as manuals using screen captures.

2.2 Screen Captures in Software Documentation

Screen captures are probably the most frequently used illustration in software manuals – a view shared by Horton (146), Houghton-Alico, and Hoft (270). For example, van der Meij and Gellevij took 100 software manuals and analysed their contents using quantitative methods. In their study they found that of the pages sampled from 100 software manuals, a vast majority (76 per cent) showed one or more screen captures, and that nearly all of the screen captures showed a whole screen or window of a program (b: 529). This is hardly a surprising result for anyone who is familiar with software documentation. Of course, screen captures are used in other areas of information design, for example in marketing where their function is the same as with all marketing material: to sell the product. Software user documents may share this function, but screen captures also have very specific roles in helping the users to find, understand, and use information.

Why are screen captures so widely used? Horton suggests that: "In technical communication, screen captures have a specific function that can improve user cognition (skill and knowledge)" (146). This idea is also expressed by van der Meij and Gellevij. In their studies, based on several empirical user analyses, they have identified four main roles of screen captures. These they named:

- switching attention
- developing a mental model of the program
- verifying screen states, and identifying
- locating window elements (b: 529)

These roles are further explained in chapter 3.

Following on this, it is logical that each role has aspects of design that affect how well the role of screen captures serves user cognition. Van der Meij and Gellevij state that each of these roles has at least one critical or essential design area: position, coverage, size, and cueing. According to van der Meij and Gellevij, some design areas can also affect more than one of the roles (b: 529). The main roles and design areas are central to designing effective software documentation.

Software documentation is somewhat different from other technical documentation. The users of software products have to divide their attention to several sources of information. In "traditional" technical documentation, the object that the user uses does not usually give back as much information as a computer. In addition, the user of a computer has specialised input devices that may give feedback. Basically, the user of a software product has three main sources of information: an input device, a manual, and a screen (van der Meij and Gellevij, b: 530). It may surprise some readers that input device is listed as a source of information, but there are input devices that give the user feedback. I will give two examples. There are computer mice that vibrate when the mouse cursor moves over a link or another important hotspot. Vibrations are used to transfer information through the sense of touch. Another example, just not as obvious, would be the small clicks of one's keyboard or the feel of the keyboard. This is feedback but we do not seem to register it consciously. This is probably why most ATMs emit beeps when you press a key – a simple keyboard that does not give the user any feedback confuses users: "Did I just press a key or not?" It is possible that humans are so accustomed to gathering information through the visual system, the part of the nervous system that enables us to see, that we often dismiss other neural impulses like auditory information or information from the sense of touch.

Understanding that the user of a software document has many information sources is important. I present these sources as a triangle; all sources are interrelated.

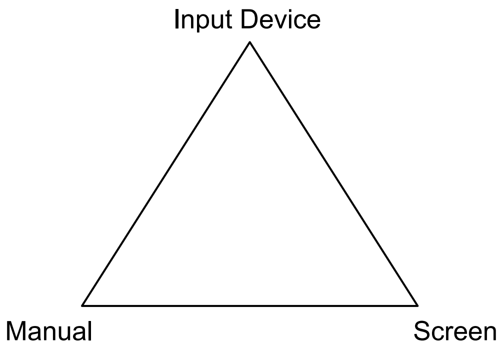


Figure 1 Users' information sources

This figure will later help me explain one of the roles that screen captures exhibit: screen captures and written instructions have an important role in switching the user's attention from one of these information sources to another (van der Meij and Gellevij, b). This role and other roles of screen captures are further discussed in chapter 3.2.

2.3 Screen Captures as Digital Information

There is one aspect of screen captures that has not yet been covered. Screen captures are digital information. This chapter helps to explain the nature of digital information on a general level, but it is important for understanding the object that is studied in this thesis. This chapter will also help me elaborate why I will not study specific media, tools, or processes to take screen captures.

I presented three different definitions of a screen capture at the beginning of chapter 2.1. Two of these definitions failed to adequately describe what a screen capture is. Why is it so difficult to

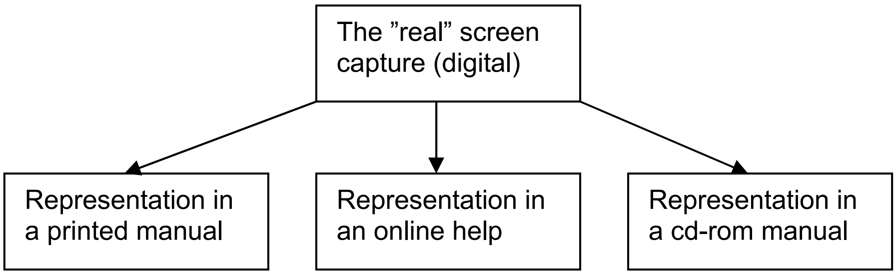


Figure 2 Digital information

This screen capture source is always digital and has all the characteristics of digital information (as opposed to analogue information):

- It can be perfectly duplicated and stored.
- It is formed from a few basic elements.
- It is independent from the media used. (Vadén)

Perfect duplication here means that copies of a screen capture are really exactly alike: they are perfect copies. The source information is also made of basic elements, two elements to be exact in the case of computers (the binary information of ones and zeroes). And finally, the information is independent from the media used. The same digital information can be represented by any number of different systems. I can best explain this by giving examples. A very long line of people sitting and standing could represent exactly the same information as ones and zeroes on your computer hard drive: standing people would represent "ones" and sitting people would represent "zeroes". Here the medium is the people standing and sitting, or more specifically the line made by the people. Analogue information, like a vinyl record used by older record players, does not have these characteristics. Each record is unique: they may look alike and sound alike, but no two records are exactly the same. The information on a vinyl record is not formed by a few basic elements, instead the information is made of thousands of different kinds of bumps and grooves on the surface of the record. The analogue information on a vinyl record is dependent on its physical representation. If you break the record, you will never be able to get a record that is exactly the same.

This difference between analogue and digital information helps to explain why specific tools or media are not as relevant for a study of screen captures. Analogue information needs a specific tool that understands that specific type of analogue information. If we continue with the vinyl record example: the vinyl record needs a record player that is specifically designed for vinyl records. Digital information however, is fundamentally different. Screen capture can be recorded by an infinite number of different digital systems (line of rocks, notes, letters, DNA, or ones and zeroes). This information can be stored, edited, and copied by an infinite number of different tools, whereas the vinyl record can only be processed with very specific tools.

In this thesis the focus is on screen captures, so the tools used to process this digital information are computer programs. There are, of course, only a finite number of programs that are actually used to process screen captures. Nevertheless, digital information can be edited in so many ways that any tool-specific research is bound to be outdated in a matter of few years. New methods, algorithms, and programs are constantly being developed. The same holds true for the media. Although paper, web and CD-ROMs (DVD-ROMs, HD-DVD ROMs, Blue-ray discs?) will probably be around for long, who can actually tell what medium is most important thirty or forty years from now? Whatever is the case, digital information will still be independent from its representation or the media used.

3 Roles and Design Areas of Screen Captures

In the previous chapter I discussed the general nature of screen captures. I tried to answer the question: "What are screen captures?" Now I aim to answer the question: "What do screen captures do?" In the van der Meij and Gellevij study the functionality and effectiveness of screen captures have been divided into two high-level items, which are explained in more detail in this chapter.

The screen capture roles and the respective design areas are studied in the framework for information design. As mentioned in the Introduction, I have chosen Carliner's three-part framework as the model that explains different aspects of design.

3.1 A Framework for Information Design

Design is more than designing the appearance of product; it also involves the underlying structure and its reception by users (Carliner 563). According to Redish, information design can be divided into two parts: the overall process of developing successful documents and the way information is presented on the page or screen (b: 163). The emphasis in this thesis is on the latter, although one could argue that the dimensions presented by Redish cannot be separated. My aim is to analyse actual documents, not to study why and how technical communicators make certain design decisions.

I have chosen Carliner's three-part framework for information design as the "whole" that includes all aspects of information design: verbal and non-verbal information. His view is needed here to show how large a number of variables there are in design, and how these different aspects work together in information design. He writes that information design is an essential ingredient for success of all technical communication. Design in technical communication should be seen as having three parts:

- Physical, helping users to find information.
- Cognitive (intellectual), helping users to understand information.

- Affective (emotional), helping users feel comfortable with the presentation of the information […] (564)

As Carliner points out, these parts are interconnected. Physical design is part of cognitive design, which is part of affective design. This is presented clearly in this figure borrowed from Carliner (565):

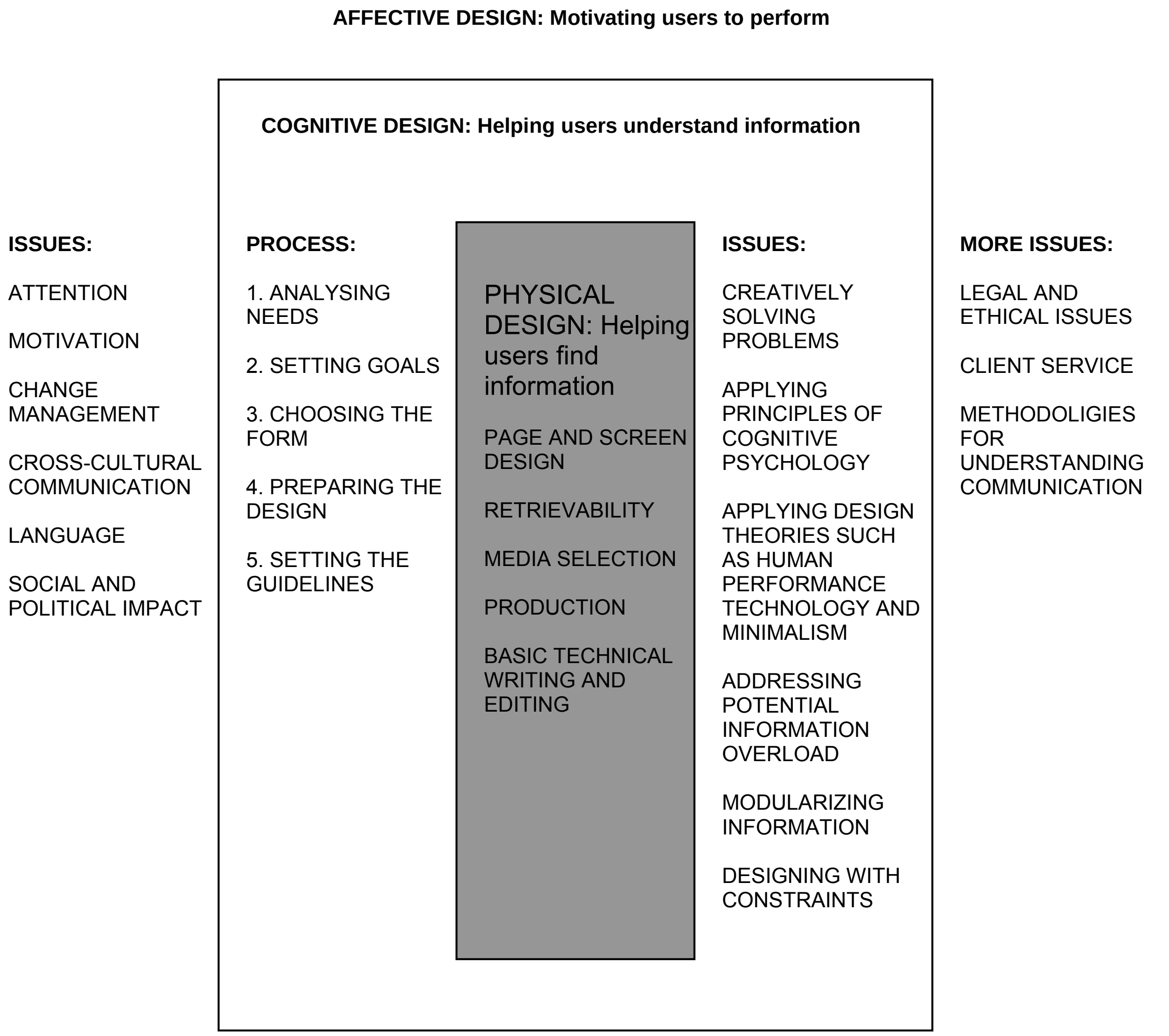


Figure 3 Physical, cognitive, and affective - A three-part model of design for technical communication products

I will study a very narrow field in the whole of technical communication, which means that some of the issues mentioned by Carliner are not as important, while others are critical: attention, motivation, and potential information overload. I emphasise information overload here because it is very easy to cause an information overload with screen captures as I will shortly demonstrate.

This emphasis on affective and cognitive design can also be seen in the work of other researchers in the field of technical communication. For example, Redish lists four critical aspects of how readers work with documents:

1. Readers decide how much attention to pay to a document.
2. Readers use documents as tools.
3. Readers actively interpret what they read.
4. Readers interpret documents in light of their own knowledge and expectations (a: 15).

Carliner does not specifically mention screen captures, but he mentions that graphical devices can be used to call readers' attention to key elements of information, and also mentions retrievability aids that help users to locate information in a document (566). His listing of retrievability aids mainly includes indexes, headers, tables of contents, etc.

Carliner's framework is flexible and can be used to show the different aspects of information design relevant for designing software documents and screen captures. He notes that although each cognitive design theory has its own definition, their goals are remarkably similar: providing users with the most appropriate information, at the exact time and place they need it (568). This is also an underlying theme that can also be found in the theory of screen capture roles. Carliner's model of information design can be used in this thesis as a high-level organisation for the different screen capture roles and design areas that are studied in the following.

3.2 Roles of Screen Captures

The roles of screen captures, as adopted from van der Meij and Gellevij, are to help the user of a software product to switch attention, develop a mental model of a program, verify screen states, and identify and locate window elements and objects (b: 529).

These roles are used to explain how screen captures can affect users. This can be done because each role has specific design areas that can be identified in the case material. The following chapters give a detailed description of screen capture roles and explain how different kinds of user support strategies can be identified in case material. In the following chapters I will address each of these four roles.

3.2.1 Switching Attention

In chapter 2.2, the user's sources of information were visualised as a triangle where each source has an interrelated relationship with the other sources of information. The role of switching attention can be best illustrated using this same figure:

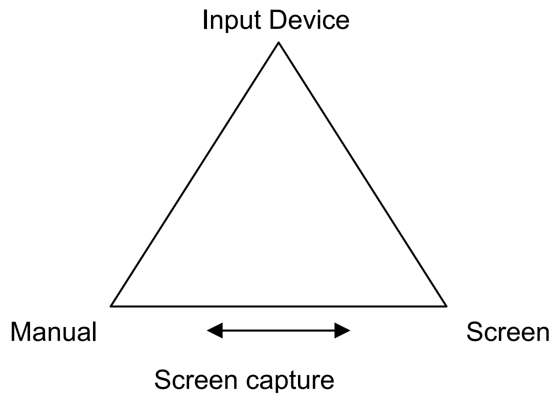


Figure 4 Switching attention between information sources

The screen capture instruction, shown on the manual-to-screen axel, helps the user to switch attention from the manual to the screen and then back again to the manual at the right moment. An efficient screen capture helps user cognition by guiding the user to the right information source at the right moment. This type of user support is evident in most software documents that contain procedures. The user is given visual cues to switch attention.

Users have to switch attention regularly from one information source to another. Van der Meij and Gellevij conclude that this is especially difficult for novice users who are more prone to err in "coordination". This can result in what they call a "nose-in-the-book syndrome", which means that the user never checks if a step presented in a manual worked properly, or if anything happened at all (b: 530). In effect, the user's demand for cognitive support is not met by the documentation, because the user is not given visual cues to switch attention.

Of course, both visual and verbal cues can be used to guide users. Van der Meij and Gellevij present conclusions from van der Meij's previous study where verbal and picture cueing was compared with verbal only cueing. Verbal-only cueing means that the visual cues, such as screen captures, were replaced with text cues such as "You should now see the Print Parameters screen" (example of the actual test instructions as presented by van der Meij and Gellevij). The results of their study were somewhat surprising: the results favoured verbal-only cueing over verbal and picture by 28 per cent versus 15 per cent (b: 532). The results of the study showed that the subjects switched attention from manual to the screen much more often if they received text-only instructions. However, it was suggested by van der Meij and Gellevij that the audience selected for this study may have significantly affected the findings, as experienced users probably need much less support for switching attention (b: 531-532). Later user studies have shown that novice users do benefit significantly from the use of screen captures (van der Meij and Gellevij, a).

How do screen captures then help users to switch attention? Van der Meij and Gellevij suggest that screen captures help the user switch attention in two ways, by:

1. Prompting the user to look at the screen at the right moment, switching attention from the manual to the screen.

2. Providing a clear point for re-entry into the manual after the user has looked at the screen. (b: 531)

Here we see how the screen capture forms a strong prompt to switch attention from the manual to the screen:

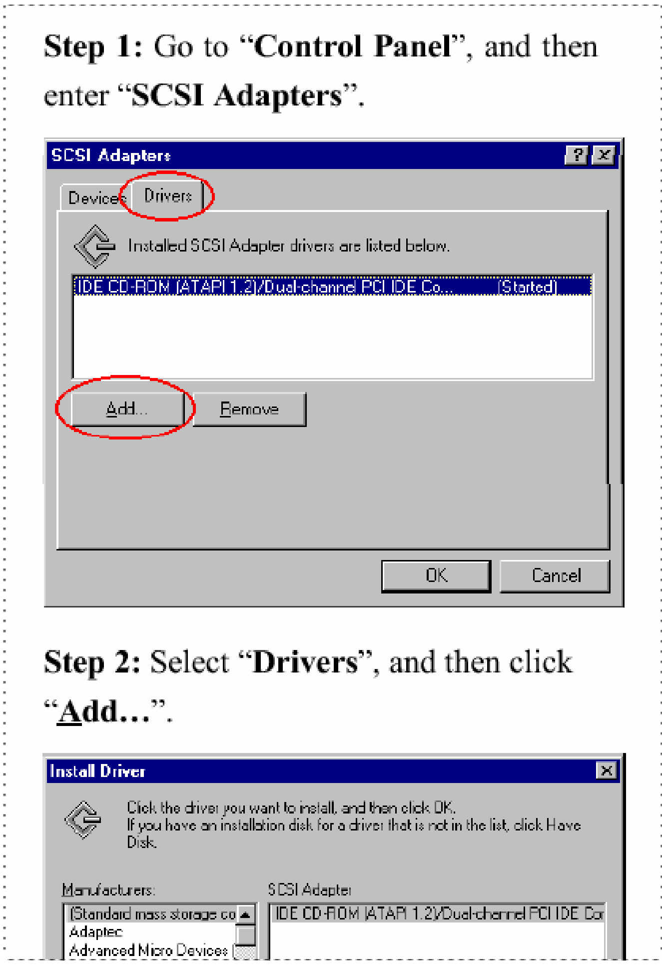


Figure 5 Switching attention (from Abit KT7/KT7-RAID User's Manual)[2]

The user is first given a set of instructions (step 1), after which the user is presented with a screen capture. This screen capture prompts the user to switch attention to the screen. The screen capture also presents the user with two added elements that are not shown on the user's screen (red circles). These added elements prompt the user to check these two specific locations.

In the example above, the numbered steps are accompanied with a picture that acts as a preview of what the user will see when looking at the screen. The red circles identify the key elements that the user should be looking at, and this also makes this a good example of how cueing can be actively

[2] The screen capture examples before chapter 5 in this study are not from the empirical material, but have been specifically chosen from select documents to illustrate certain situations and concepts. The source of each screen capture is indicated in the caption of the figure. Screen captures shown in the analysis part are from the empirical material.

(Note that the screen captures presented in the figures of this thesis are in most cases screen captures of screen captures, which means that the quality of the figure cannot be affected if the original screen capture has a poor resolution.)

used to help user cognition. The red circles form a logical path for the user, from top to bottom: drivers -> Add -> re-entry to the manual. Screen captures also prompt the user to switch attention from the screen back to the manual. After seeing a similar window on the screen, the user is more likely to switch attention back to the manual.

A critical design area for switching attention is positioning (van der Meij and Gellevij, b: 531) which will be introduced in chapter 3.3.1.

3.2.2 Developing a Mental Model of the Program

Creating a mental model is most critical in problem solving and in new situations. Mental models and the user's cognition help to detect, define, diagnose, and solve problems (van der Meij and Gellevij, b: 532).

The user of a software product needs to understand the general structure of a program or its user interface to effectively learn to use the program. Van der Meij and Gellevij conclude that users of a software product become acquainted with its structure by creating a mental model of the program. A mental model is created by getting familiar with "the standard layout of the windows in a program" (b: 531). This makes it important for the technical communicator to choose the right content for software documentation, which is an aspect also presented by Redish in the context of the information design process (b: 163).

The definition of this role seems to imply that the process is passive. Users do not necessarily actively analyse the components of the program; the mental model is created unconsciously. This impression is reinforced by a list by van der Meij and Gellevij that shows how screen captures contribute to the development of a mental model:

1. Screen captures acquaint the user with the main windows.
2. Screen captures explain the spatial layout of a window.
3. Screen captures develop a sense of logical flow, or progression, of windows. (b: 531)

Acquainting the user with main window types means that when a screen capture appears in instruction, it acts as a strong prompt for the user to analyse the window. The user can also be presented with other visual cues that identify important elements in windows (b: 532). For example, such cues can be added or active elements used in windows.

Screen captures explain the spatial layout of a window. A screen capture can be used to actively help the users to understand a spatial layout of windows. This means that a screen capture can, for example, show where a pop-up window appears when the user performs a step or part of the instructions (b: 532). The following figure, Figure 6, illustrates how a screen capture can be used to introduce the layout of windows to the user. The screen capture acts here as a strong visual cue for the user. It tells the user where the next menu will appear, and what the relevant items are in that menu. The screen capture shows only a small part of the screen here; the user's attention is focused in the relevant part of the screen.

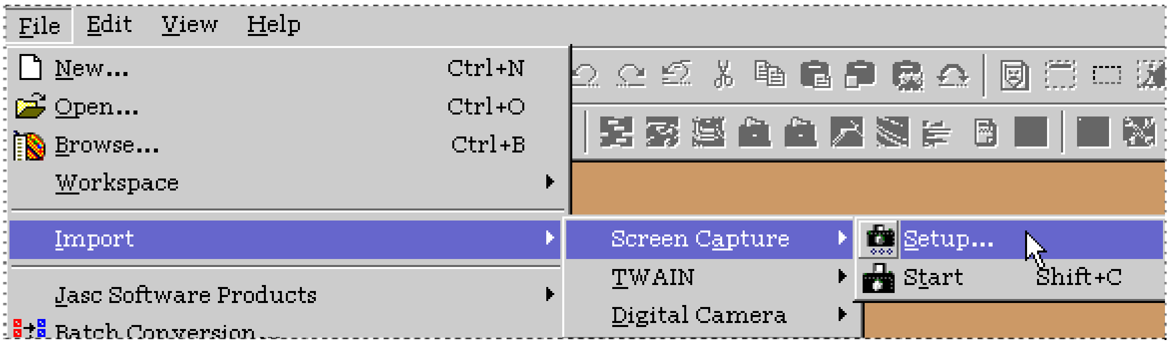


Figure 6 Spatial layout and logical flow (from Disdat Design Manual)

Here the window types and layout are also introduced in a sequence similar to a real procedure: Import → Screen Capture → Setup. This "context of work", according to van der Meij and Gellevij, is beneficial to the user (b: 532). The example was taken from Disdat Design Manual where it is used to show the user how to set up and import a screen capture to a program. I have omitted the accompanying text instructions in order to focus the attention to this specific role of screen captures.

Screen captures develop a sense of logical flow of windows. The figure above also serves as an example of another of the three sub-roles of developing a mental model. In addition to introducing the layout of windows, it shows the user the logical flow of windows. Only one screen capture is needed here to explain a user the following: "To access the setup menu for importing screen captures, open the File menu, choose Import, choose Screen Capture, and then click Setup." This screen capture works well because it gives the user the right coverage. It does not show too much, but conveys the logical flow of the program. A series of screen captures can be used to give the user a sense of logical flow of windows (van der Meij and Gellevij, b: 532-533). This is also illustrated in Figure 5 above where the steps used in the manual catch the logical flow of windows: "first you will be presented by this window and when you perform this step you will be presented with this window."

Coverage and size are the design areas critical for creating a mental model of a program. These design areas will be explained in detail in chapters 3.3.2 and 3.3.3 respectively.

3.2.3 Verifying Screen States

Horton proposes that screen captures should mainly be used in two situations: when the screen is not visible and when the user must verify the display (146). The latter does coincide with screen captures' role of verifying screen states. This role is especially important for novice users who often want to know that they have followed instructions correctly (van der Meij and Gellevij, b: 535), and thus this role supports user motivation.

Verifying screen states is also important for more advanced users. Advanced users often want to find the information they need fast and then return to their work (Redish, a). What van der Meij and Gellevij suggest is that users who read a software document for quick reference can also benefit from screen captures. They present that there are at least these two situations where a user benefits from the role of verifying screen states:

4. When the user is presented with an error message or a warning,
5. When the user is at the beginning of a procedure (b: 535-536)

When users are presented with an error message or warning they turn to documentation for help, but often the users have difficulties in finding relevant information. Van der Meij and Gellevij presented a study where van der Meij had inventoried 60 manuals: the study showed that 86 per cent of the manuals did not use any means to highlight such problem-solving information, and an obvious correction to this can be the use of screen capture to catch the user's eye (b: 535). In this way the users would be able to see at a glance that the page they are looking at contains information about solving an error message. This kind of information design is important for the user's ability to find the information needed (Carliner 564). I do not think this means that there should be a screen capture of every single error message. Problem solving information could possibly be highlighted by using a screen capture that the user can identify as an error message.

The other situation where verifying screen states is of importance is at the beginning of a procedure. Van der Meij and Gellevij explain that users can benefit if they see the manual text accompanied with a picture that shows the screen state that should be present before beginning the task. They mention other conditions as well, such as verifying that the correct file has been loaded, or that the program is in the correct mode (b: 536). I have noted that some software documents often use such "preview" pictures, and usually these pictures are used to name the elements of the window or screen.

How do screen captures help the users to verify screen states? According to van der Meij and Gellevij, screen captures:

1. Provide a visual entry point into a manual when user is skimming the manual.
2. Support progress checks. (b: 536)

Skimming is a very common user action in problem situations, and screen captures can offer important assistance. Redish notes that users skim and skip, they read just enough to find the information they need. She mentions that helping users to find what they need quickly is of critical importance in technical documents; if this need is not satisfied, users get frustrated (a: 3). Screen captures provide an entry point into the right information by showing a preview picture that can be identified by the user.

Screen captures also support progress checks. This example shows how a screen capture is used to mark the reader's progress:

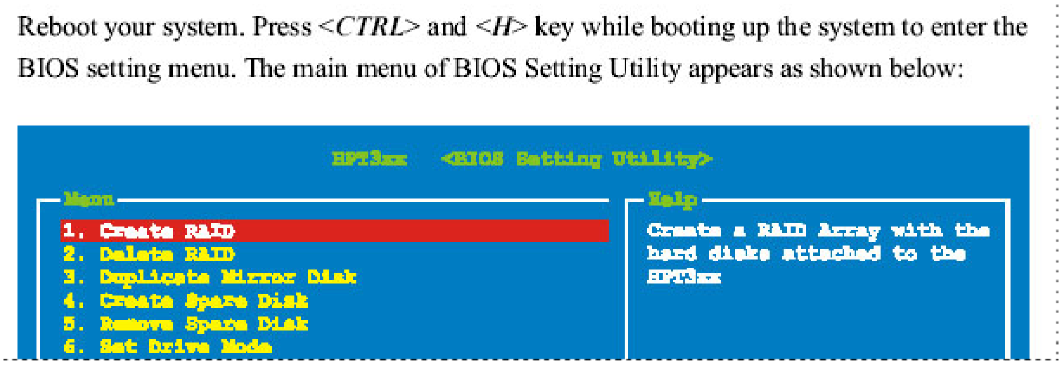

Reboot your system. Press <*CTRL*> and <*H*> key while booting up the system to enter the BIOS setting menu. The main menu of BIOS Setting Utility appears as shown below:

Figure 7 Verifying screen states: supporting progress checks (from Abit KT7/KT7-RAID User's Manual)

The user is not shown a "before picture" at the beginning of the instructions because it is not needed. The user is first given information that should show up on the screen. The text provides a verbal cue for switching attention to the screen, and the screen capture acts as a non-verbal cue. The user verifies the screen state, and the correct screen on the user's display prompts the user to switch attention back to the manual.

This is one way to help the user verify screen states, but a step in a procedure could be presented with two pictures, "before" and "after". This can be done only if the changes in the screen state are easy to see at a glance. In most situations such minimalist strategy is not used because most screen captures look a lot like each other. Even if cueing was used (see chapter 3.3.4), it would be difficult for the user to see the difference. The lack of verbal cues would probably negatively influence the effectiveness of switching attention and verifying screen states. Using only screen captures may work if the coverage is right.

Coverage is the critical design area for verifying screen states, and it will be presented in more detail in chapter 3.3.2.

3.2.4 Identifying and Locating Window Elements and Objects

Most user interfaces have a lot of different objects, windows, icons, etc. This is also shown in the material chosen for this thesis. Identifying the correct elements can be difficult, especially for novice users. Van der Meij and Gellevij give an excellent example: they counted all the possible elements (menu-options, icons, and symbols) that could be manipulated on a Microsoft Word start screen. They arrived at a figure of 63 options (b: 537).

I decided to do a similar test on the most recent version of Microsoft Word, Microsoft Word XP. The result was amazing: 256 objects that could be moved or clicked. I did not even count all the possible sub menus or toolbars that can be activated because van der Meij and Gellevij did not specify that they had included these in their total. I collected most of these objects into a single toolbar and the results can be seen below. The figure showing all Windows XP toolbars is an exaggeration, but it shows how many elements a modern user interface can have.

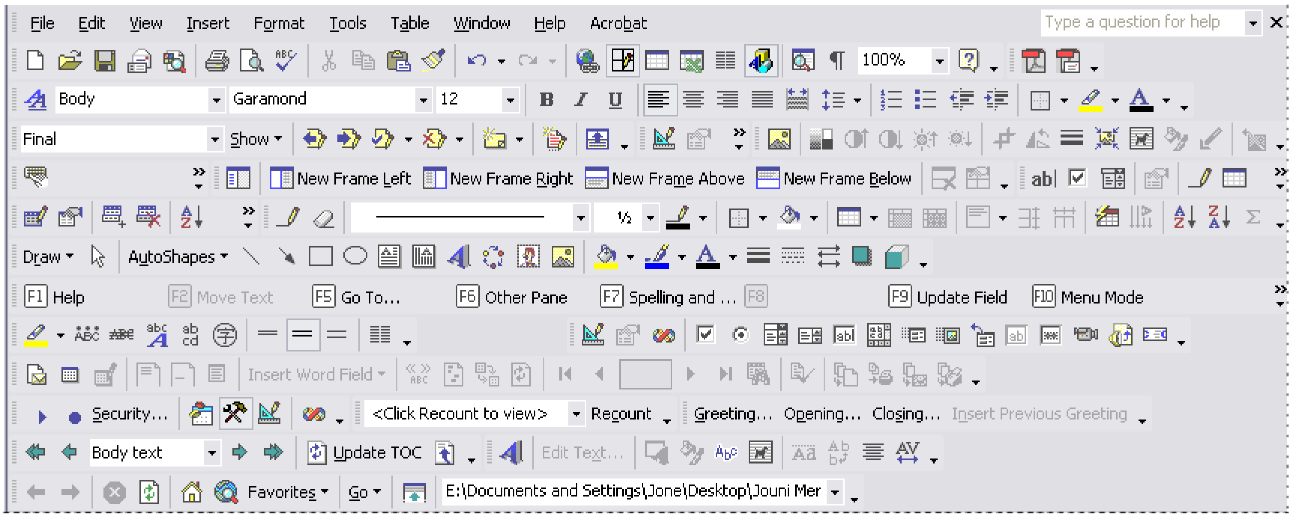


Figure 8 Why identifying and locating is important in technical communication (from Microsoft Word)

The need for identifying and locating the correct elements seems to have grown; at least this is the case with Microsoft Word. I presume that most software products would yield similar results because newer versions of software often include more options and functions. In addition, screen resolutions and monitor sizes have constantly increased throughout the history of personal computing, making it possible to display more elements and icons on computer screens.

Van der Meij and Gellevij propose that screen captures can reduce task complexity and underutilisation[3] by:

- Identifying window elements and objects

[3] Underutilisation means that most users use only a fraction of the commands and functions available in a software product. Van der Meij and Gellevij give several examples of this (b: 538).

- Locating window elements and objects (b: 538)

Screen captures can help users to identify the correct window elements and objects, as we saw in Figure 5 above. In that picture we see how red circles have been added to the screen capture to guide the user. Users searching for a specific tool can benefit from this type of screen captures (van der Meij and Gellevij, b: 538). I call these 'added elements'. It is a term that covers all types of elements that have been added to a screen capture to guide the user (numbers, circles, arrows, lines, colours, etc.). These elements are usually very easy to spot and this is why added elements are one of the important aspects analysed in the analysis part of this thesis. This type of cueing is a very easy way to motivate the user to perform an action.

Cueing is the design area that is critical for the role of identifying and locating (b: 538), which will be introduced in chapter 3.3.4.

3.3 Design Areas of Screen Captures

The four roles of screen captures, described in the previous chapters, can be used to analyse screen captures in the case material. To analyse the material chosen for this thesis, I need to have some means of identifying different screen capture roles. This can be done by analysing different aspects of screen capture design. These are what van der Meij and Gellevij call design areas.

The main design areas of screen captures are: positioning, coverage, size, and cueing (b: 529). Each of these design areas is critical for at least one the roles of screen capture:

Table 2 Roles of screen captures and their design areas

Role	Design Area(s)
Switching attention	→ Positioning
Developing a mental model of the program	→ Coverage, size
Verifying screen states	→ Coverage
Identifying and locating window elements and objects	→ Cueing

The table above names the main roles and their respective design areas. It should be noted that one of the roles, developing a mental model of the program, has two critical design areas.

The following table, adopted from van der Meij, shows the same design areas, but gives short descriptions of each design area. I have reorganised the table to better reflect the organisation of this thesis, but the content is the same:

Table 3 Description of design areas and their criteria (van der Meij, a: 1)

Design dimensions [design areas]	**Description**	**Criteria**
Positioning	Place of the picture in relation to the text	- use of column(s) - redundancy
Coverage	What is shown on the user's actual screen	- (non) use of context
Size	Percentage reduction of actual screen	- continuation - legibility
Cueing	Visual signal for drawing the user's attention	- presence/absence - naturalness - task relevance

This table shows what van der Meij means with each of the design areas. The table also lists the criteria for the design areas. For example, criteria for cueing present both the presence and absence of cueing, because both are relevant for this design area.

3.3.1 Positioning

Positioning is defined by van der Meij and Gellevij in the following way:

> Positioning designates the placement of text and screen captures in relation to one another, reflecting their relationship or independence. 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. (b: 540)

Positioning is a critical design area in switching users' attention (b: 531). Positioning a screen capture in the right place in an instruction gives the user a visual cue that helps the user to switch attention from manual to the screen at the right moment.

The following example, Figure 9, shows the positioning of a small screen capture element:

Figure 9 Positioning, example 1 (from Philips CustoMax 4 manual)

The screen capture here is an integral part of the information given to the user. The symbol for forward is not a part of an instruction here; it only serves to introduce an element.

Presenting a screen capture as part of a direct instruction is important for user cognition by associating the screen capture with a specific task. This strategy is far more common in technical communication and is shown in the following example. The figure shown here is a direct instruction from a manual of Macromedia's a web-authoring tool:

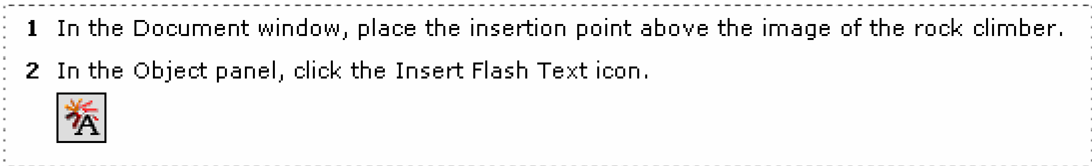


Figure 10 Positioning, example 2

In this figure we see a screen capture of a user interface element (button or an icon) with direct instructions. The icon is not actually needed for the user to carry out the instruction, but it is an important cue for switching the user's attention and identifying an object. Here the positioning of the Insert Flash Text icon could be improved by inserting it into the actual text:

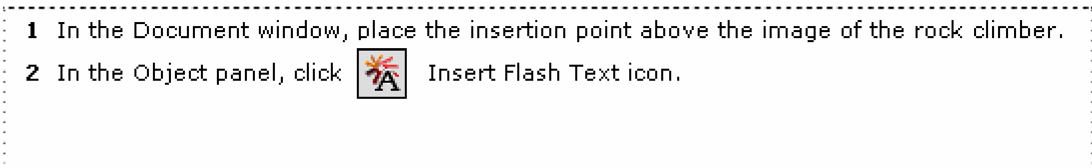


Figure 11 **Screen capture as an integral part of an instruction [edited from the previous figure]**

Here I inserted the single screen capture of an icon (Insert Flash Text) into the actual instruction to act. In this way the user cannot ignore or fail to notice the screen capture. In the figure above the user receives a much stronger cue to switch attention from the manual to the screen. The positive effects of dual coding, presented in chapter 1, are taken into account here.

In a similar situation van der Meij and Gellevij conclude that a screen capture appearing after a verbal instruction can be overlooked by the user, because it is not an integral part of the instruction to act (b: 531). It is logical that a visual cue that is part of the instruction gives a stronger cue to switch attention. Of course, there are other considerations that come into play. It would be too simplified to assume that screen captures should always form a part of the actual instruction to act. This is underlined by the results of user studies with text-only vs. text and picture instructions, as introduced in 3.2.1. The study does suggest that the previous experience of the user is an important factor and should always be taken into account. Expert users might not need as much visual support as novice users.

3.3.2 Coverage

Coverage is defined by van der Meij and Gellevij in the following way:

> Coverage refers to the desktop, window, window element, or object displayed in the screen capture. One extreme position on this dimension is screen capture that shows the full screen. The other extreme position is a screen capture that presents a single object (icon, symbol, menu option) that the user can manipulate with an input device. (b: 540)

Coverage is a design area critical for developing a mental model of a program. It is also important for verifying screen states. This is the only design area that, according to van der Meij and Gellevij, is critical for more than one of the roles of screen captures (b: 533).

Coverage can be best explained using a picture:

Figure 12 Coverage (from Windows 98 main window)

Here the screen capture covers a large area of the users display. It shows the whole screen, not just one part of the screen or one window. This screen capture from an actual software document is of poor quality, but if you need to develop a mental model of a complicated program (here the operating system), you need a larger coverage. For example, screen capture in Figure 9 does not tell the user what the actual software looks like and how it might operate, because the coverage of the screen capture is too small.

The screen capture shown above helps to explain the internal logic of the software. Most of us are familiar with the software product displayed here, so the example might not be the best possible. If we did not know the software, this screen capture and accompanying text instructions would be very important for us. They could be used to give us a mental model of the software.

Coverage is also critical for another role of screen captures: verifying screen states. Usually this role helps the user to verify his actions by presenting a screen capture that shows what the user should see after a correctly completed action. If this "after" screen capture is almost identical to the situation before the action, the user will have difficulties discerning between these pictures.

The solution is to adjust coverage so that only the relevant items are shown. Van der Meij and Gellevij mention that correct coverage helps the user because the relevant information is focal in the screen capture (b: 536).

3.3.3 Size

Size is defined by van der Meij and Gellevij in the following way:

> Size refers to the reduction rate of the desktop, window, or object in the screen capture compared with its actual screen size. (b: 540)

Size is important for creating a mental model of a program (van der Meij and Gellevij, b: 533). In this context I would define size as a relation between the screen capture and the original screen element, and as size in relation to the page in a software manual.

Horton mentions size but his focus is mainly on the negative effects of screen capture size. His concern is that large screen captures can make software documents larger and longer (146). It is true that this can be a problem. I have seen documents that have a small introductory chapter followed by five pages of huge screen captures. The user of such a manual cannot follow the procedures outlined in the manual because the sense of connectedness is lost. Van der Meij and Gellevij have made similar observations (b: 534). Consider the screen capture shown in Figure 12. If this screen capture would not have been reduced in size, it would now span an entire page. It would probably show different elements of the software better, but a large picture would also make it harder for the user to follow the accompanying text instructions. If this same picture were to be used in a procedural instruction, the steps of the procedure would be too far apart for the user to develop a mental model of the program.

3.3.4 Cueing

Cueing is defined by van der Meij and Gellevij in the following way:

> Cueing refers to the presence or absence of signalling techniques that draw the users' attention to relevant window element or objects Signalling techniques include the use of (colored) hairlines, circles, callouts, and blurring. (b: 540)

According to van der Meij and Gellevij, cueing is a design area critical for identifying and locating (b: 538). This role was illustrated in Figure 5. The user was given very clear cues (red circles) that were used to identify and locate important elements for the user. The user was first given an instruction and then a screen capture showed what the elements look like and where the elements mentioned in the instruction are situated.

If no cueing is used the user may have to exert too much effort to finding the correct elements:

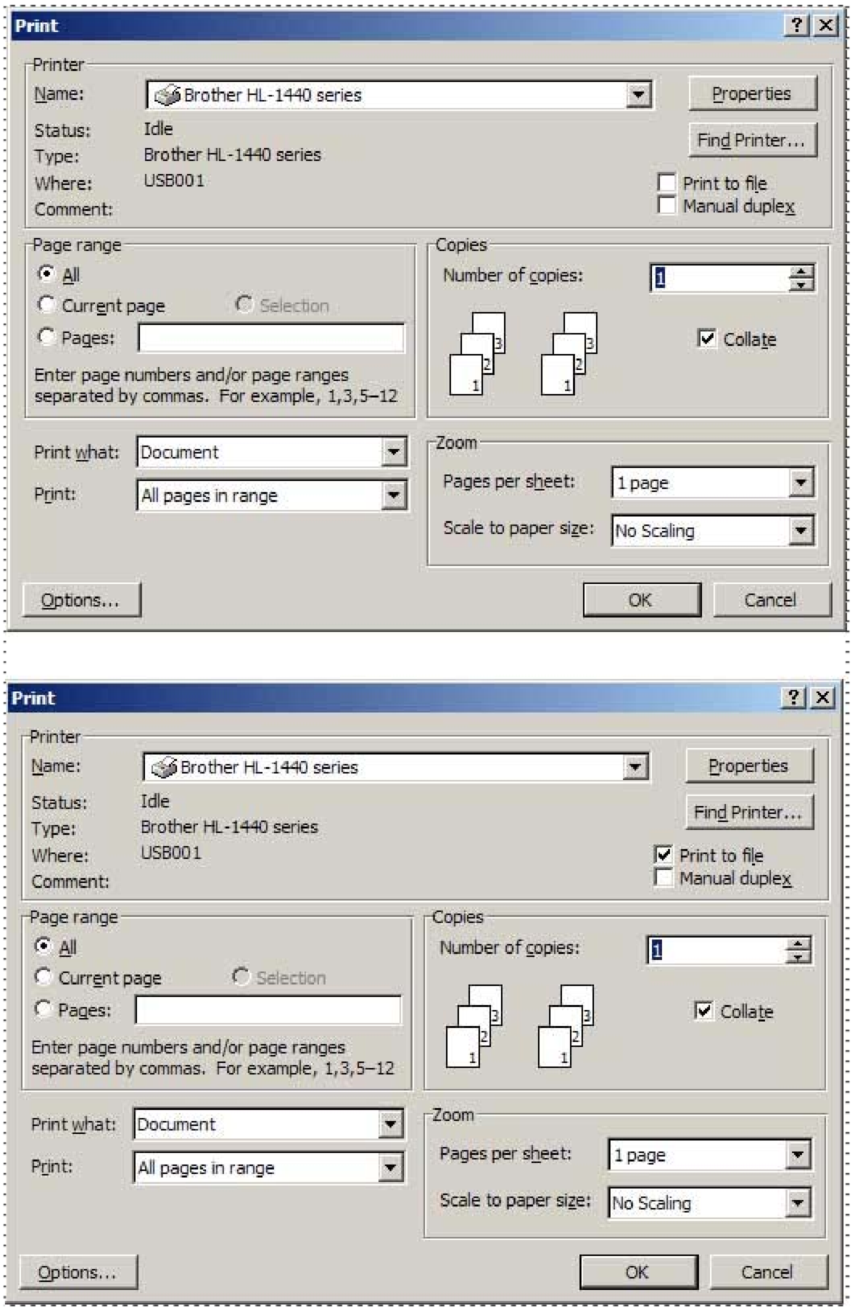


Figure 13 Picture "before" and "after" without cueing (from a Microsoft Windows print dialog)

Here the user is presented with two screen captures that show the screen state before and after an action. Without cueing the user will have real difficulties finding the relevant information. In effect, the technical communicator's attempt to aid the user has become a "spot the differences" game. The change in the figure above was really small (ticked "Print to file"), but it has been noted in cognitive psychology that even large changes in a visual scene can be missed by humans. For more information on this phenomenon called change blindness, see for example Becker and Pashler.

4 Material and Methods

Instead of analysing all screen captures in one or two documents, this study was conducted by selecting random screen captures from a larger set of documents. The material of this study comprises of 50 screen captures taken from five software documents.

The software documents were chosen because they are freely available on the Internet, and the chosen software applications are also fairly common - they are not "professional", highly-specialised tools. The table below lists the chosen documents and a brief description of each software application:

Table 4 List of the software documentation studied

Name of the software document	A short description of the software
Adobe Reader 8 User Guide	A common PDF reader.
Ad-Aware 2007 User Manual	An anti-spyware program.
Foxit Reader 2.0 User's Manual	A PDF reader.
Nero 6 Ultra Edition QuickStart Guide	CD/DVD authoring software.
TreePad Lite 2.9.5 User Guide	A personal information manager.

The software applications chosen are not very specialised, but they have a lot of features. It was possible to include these complex products because I analyse a small set of screen captures in each case.

The Adobe Reader 8 User Guide is a 148-page software document that contains a lot of both descriptive information (user interface descriptions, descriptions of the features, etc.) and procedural information (step-to-step instructions on how to accomplish tasks). It contains hundreds of pictures, and a vast majority of the pictures found in this document are screen captures.

The Ad-Aware 2007 User Manual is 60-page software document. It has less procedural information and concentrates more on describing the functionality and what the user can do. Screen captures are the predominant type of picture here.

The Foxit Reader 2.0 Manual is an 88-page software document. It contains roughly 200 screen captures in all. Again, screen captures are practically the only type of picture used in the document.

The Nero 6 Ultra Edition QuickStart Guide has both procedural and descriptive information. For a quick start guide it is quite long: 46 pages. Almost all pictures are screen captures.

The TreePad Lite 2.9.5 User Guide is a 30-page software document similar to the Adobe Reader 8 software document. It contains 24 pictures, of which 23 are screen captures.

The screen captures were chosen by first selecting a random page using a random number generator[4] and then choosing the first screen capture encountered on the page. I took a screen capture of the page containing the screen capture to be analysed, and numbered the figure consecutively and added the document name and the page where the screen capture was found. (The page numbering of the figures follows the conventions used in the material.) For example, the first screen capture is named: Picture 1 – Adobe – page 23. I will use this convention when referring to the material in the full list of screen captures found in Appendix I.

I will give several examples from these software documents, but it should be noted that many specific examples of the roles and design areas have already been presented in the previous chapters of this thesis, and I will sometimes refer to these examples that are already more familiar to the reader.

For the analysis part of this thesis I needed a way to classify screen captures. The criteria given by van der Meij and Gellevij and presented in Table 3 are a very good starting point. The

[4] Generating "true" random numbers is actually not as easy as it would intuitively seem. This is because computer programs create numbers in a predictable fashion using mathematical formulas. The random page numbers used in this thesis were generated by using a free random number generator available at http://www.random.org/, which generates true random numbers from atmospheric noise. A page number range was set first in the Integer Generator, for example 1 to 65 pages. Then random page numbers were generated until there were enough pages that contained a screen capture. For more information on randomness, see *Introduction to Randomness and Random Numbers* (Available in http://www.random.org/randomness/), and for more information how Random.org has been used for random sampling related for scientific studies, see *Testimonials - Random Sampling* (available in http://www.random.org/testimonials/sampling/).

characteristics of the screen capture design areas can be used to analyse screen captures because they share the features share these features:

- The identifiers can be easily spotted and are in most cases unambiguous.
- Analysis of these identifiers does not require the study of users, or an excessive amount of work or raw computational power, since I do not have access to usability laboratories or other resources.
- These identifiers should be universal. Characteristics can be found throughout the material, not just in a specific case. Analysing these should not extract too case-specific information. Although this requirement should be met by the material chosen, it is likely that this can only be confirmed when the actual analysis is conducted.

The analysis will consist of going through each screen capture chosen and checking in that screen capture if:

- each of the design areas is relevant in the particular case.
- the design areas are used in a way that accomplishes one of the three aspects of information design.

When the analysis has been recorded and studied I will present my findings for each screen capture role. These results and a closer analysis of differing screen captures will be used to find out if screen captures are used to accomplish the three aspects of information design: physical (helping users find information), cognitive (helping users understand information), and affective (motivating users). The analysis should help find out if screen captures are used in a way that benefits the user, and it should illuminate what the potential problems in the material are. When this part of the analysis is ready it should be possible to map the results in Carliner's model and list which aspects of information design are supported and which ones lack support. The individual design areas will be analysed as follows:

Positioning. Either a screen capture is an integral part of an instruction, or it comes after the instruction. Each screen capture will be analysed if it is used as an integral part of an instruction.

This design area should be fairly easy to spot in the screen captures, and the analysis of positioning should reveal if the screen captures are properly used to switch the user's attention.

Coverage. As coverage is a design area that critical for two different screen capture roles, each screen capture is analysed separately for this design area, and once for each role. For developing a mental model, the screen capture should give sufficient coverage of the application. Noting this is probably more difficult than the analysis of positioning, but basically it means analysing if the screen capture is too small and does not give the coverage that would enable the user to understand what the logical flow of action is. For verifying screen states, the screen capture should show how a completed action has affected the application, and the coverage should be such that the relevant effect on the application is shown. This aspect of coverage will be easier to analyse in the material.

Size. The size of the screen captures in relation to the text (or software document page) will be analysed. If the screen captures have not been reduced in size, and cause, for example, procedure steps to become too disconnected, the screen capture does not help the user to develop a mental model of the program. This part of the analysis also requires looking at the surrounding text, not just the screen capture.

Cueing. This design area, if correctly applied in the screen capture, is very easy to spot. Either there is an added element (some visual aid added to the screen capture by the writer of the software document) or an active element (an element of the screen capture such as a field or a button). Also, if there is an active element in the screen capture, it should be the correct one. For example:

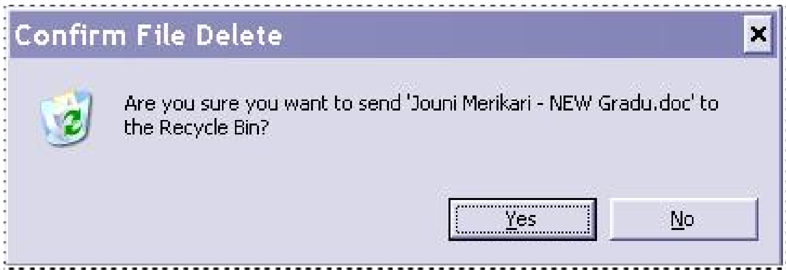


Figure 14 Active elements (from a common Windows XP dialog window, not part of the material)

In the figure above, the button labelled as 'Yes' is the active element. If the instruction accompanying this screen capture would be to confirm the deletion by clicking 'Yes', then the active element in the above figure would be used in a way that helps the user to identify and locate window elements. On the other hand, if the user was instructed (for some reason) to click 'No', then this screen capture would be misleading as the wrong window element is active. Active elements can be used as cues, so in the figure above the user's attention will centre on the highlighted text. Technical communicators can use this to their advantage when designing instructions, and this is easy because there is no need to add more elements or edit the picture. However, this characteristic can also work against the technical communicator if it is not used consciously: an active element that contradicts verbal instructions can mislead and confuse the user.

5 Results

The results of the analysis are presented here for each screen capture role. After each role has been presented I will present the results in the context of Carliner's three-part framework for information design. A detailed list of all the findings is given in the table in Appendix I.

Each design area was analysed once. The exception was coverage that was analysed twice, because coverage supports two different screen capture roles. This meant that there were altogether 300 elements that needed to be analysed – and this was only for five documents and ten screen captures per document. This verified that it had been a good decision to analyse a smaller number of screen captures from a larger set of software documents, instead of analysing all screen captures from a couple of software documents. Fully analysing any two documents of the documents chosen for this thesis could have meant analysing hundreds of screen captures, and thus most likely well above a thousand different elements to analyse. This would have been a too extensive analysis with the resources and time available, and also taking into consideration that this is master's thesis. Admittedly, a larger set would have resulted in more accuracy, but the smaller set chosen for this study was sufficient for analysing design areas and their relationship with the three aspects of information design.

During the analysis it was noted that analysing all design areas for each screen capture was not possible, or, more accurately, not all screen capture design areas were relevant in all of the screen captures studied. Because of the way my material was chosen, some of the screen captures used were not a part of an instruction or procedure at all. Most of the screen capture roles presented by van der Meij and Gellevij are better describe users' interaction with procedural information. Positioning was the first area studied and it was soon apparent that positioning could not be studied for screen captures that were not part of a procedure. As the screen capture roles for switching attention, verifying screen states, and also developing a mental model of a program, are related to user actions and motivating the user to act, these descriptive screen captures could not be analysed in any meaningful way for the respective design areas, and were marked as irrelevant.

The analysis was carried out as planned, but all cases where the design area was not relevant for the particular screen capture were marked and counted as irrelevant. The following is an example of a typical screen capture that was not part of a procedure:

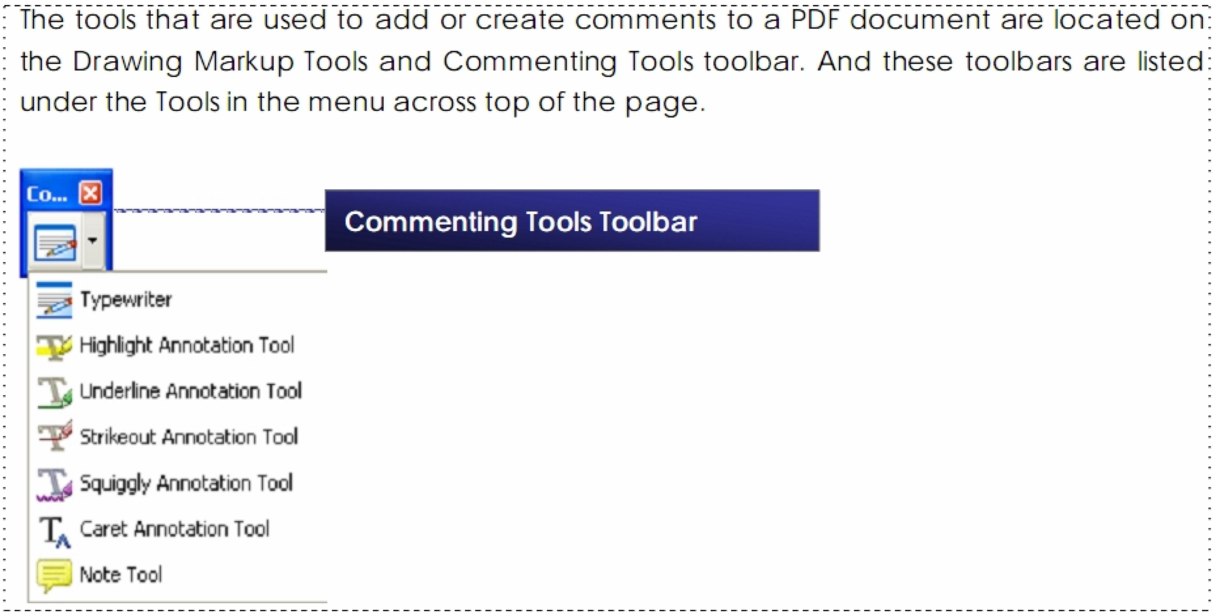


Figure 15 'Irrelevant' screen capture that is not part of a procedure (from Foxit Reader 2.0 User's Manual, screen capture 23 in the material)

The screen capture above gives descriptive information to the user that is not related to any specific instruction or procedure.

Some screen captures, most notably all of the screen captures in the Ad-Aware 2007 User Manual, had to be analysed twice, because it was not always clear in what context the original screen capture had been used. The document was quite interesting because some of the screen captures that were used in a descriptive role, i.e. to identify elements in the program, also contained added text that looked like it belonged to a specific procedure. Although I also studied the relationship of text and figures, this re-analysis was required in cases where it was not obvious if the original screen capture had been used in a procedural instruction or not. Below is an example of screen capture from the document:

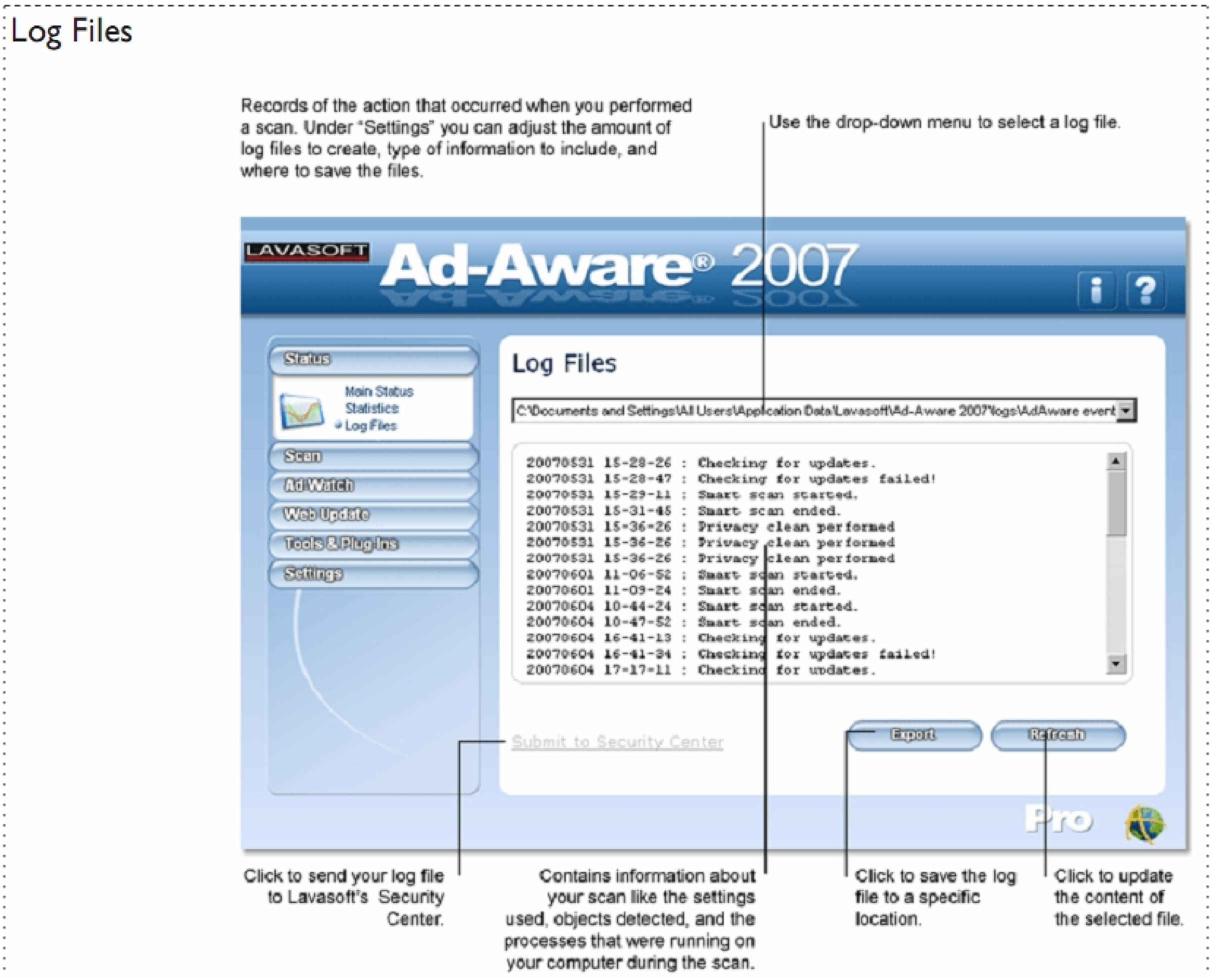


Figure 16 Added "instructions" in a screen capture that is not part of any instruction (from Ad-Aware 2007 User Manual, screen capture 14 in the material)

The figure above described a window called 'Log Files', but it also contained several added text elements that prompted the user to follow an action. In addition, these small "instructions" were not presented as part of a procedure and were not necessarily related to each other in any way.

The results will be described in the sub-chapters below. As the titles suggest, I have chosen to describe the effectiveness of the screen capture roles through their respective design areas. There are two reasons for this structure. First, analysing the effectiveness of the screen capture roles is the true focus of the analysis, which is only expressed through the design areas. And second, there can be several design areas that affect one role. Describing the changes per design area could be confusing as the chapter related to coverage would include findings for two different roles.

5.1 Switching Attention (Analysis of Positioning)

Positioning was easy to analyse in the material, but it was only relevant for screen captures used as a part of an instruction or procedure. Out of 50 screen captures, the following number of screen captures were:

- using positioning in a way that supported the user: 14
- not using positioning in a way that supported the user: 16
- irrelevant: 20

There were major differences depending on the document studied. Two documents had almost all of the screen captures that supported switching attention, and two other documents had almost all of the screen captures that did not support switching attention. The remaining document had a very large number of the irrelevant screen captures. The table below shows the breakdown of findings per document.

Table 5 Results for positioning per document

Name of the document	Uses positioning to support the user	Does not use positioning to support the user	Irrelevant
Adobe Reader 8 User Guide	7	-	3
Ad-Aware 2007 User Manual	-	1	9
Foxit Reader 2.0 User’s Manual	6	2	2
Nero 6 Ultra Edition QuickStart Guide	-	7	3
TreePad Lite 2.9.5 User Guide	1	6	3
Total	**14**	**16**	**20**

These results could not be used to make any generalisations about how switching attention in supported software documentation, other than that there were whole documents where the technical communicator has chosen an information design strategy that either supported or did not support switching attention.

The following is an example from the Nero 6 Ultra Edition QuickStart Guide where switching attention could have been supported by adding the icon into the actual procedure, instead of giving a separate screen capture:

3. Select the desired files and click on the 'Add' button to transfer them to the compilation.

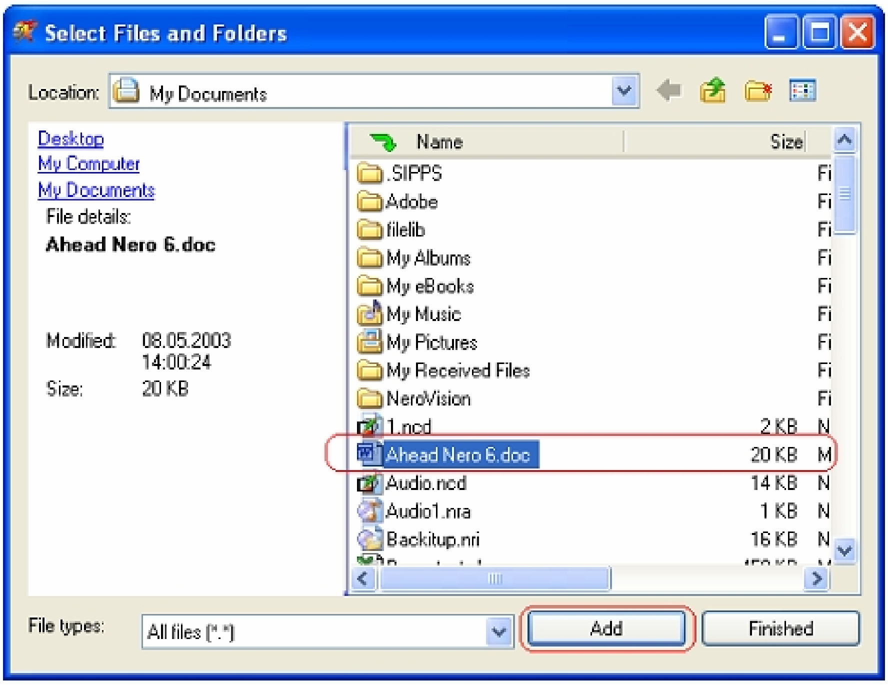


Figure 17 Switching attention not supported (from Nero 6 Ultra Edition QuickStart Guide, screen capture 35 in the material)

There were a lot of screen captures like this that could have supported switching attention better, so it will be possible to use these examples to formulate better guidelines for information design, as discussed in the conclusions part of the thesis.

5.2 Developing a Mental Model of a Program (Analysis of Coverage and Size)

Size and coverage were fairly easy to spot from the material. As for coverage, in most cases it was easy to say if the screen capture gave enough coverage for the user to develop a mental model of the program being used. This design area was analysed both for developing a mental model and for verifying screen states. Unlike verifying screen states, developing a mental model had no irrelevant screen captures. Although both of these are affected by the design are of coverage, only verifying screen states is strictly related to procedural information (by showing how an action has affected the program). Descriptive screen captures with the correct coverage can help the user create a mental model of the program.

Out of 50 screen captures, the following number of screen captures were:

- using coverage in a way that supported the user: 25
- not using coverage in a way that supported the user: 25
- irrelevant: none

Although the findings were split 50:50, once again the results per document were more interesting as there were two documents (TreePad Lite 2.9.5 User Guide Foxit Reader 2.0 User's Manual) in particular that were not using coverage very well to develop the mental model of the user. The rest of the documents managed this design area better, but there seems to be more issues with consistent use of coverage, if compared with the results for size and positioning. The table below shows the breakdown of findings per document.

Table 6 Results for coverage (developing a mental model) per document

Name of the document	**Uses coverage to support the user**	**Does not use coverage to support the user**	**Irrelevant**
Adobe Reader 8 User Guide	8	2	-
Ad-Aware 2007 User Manual	7	3	-
Foxit Reader 2.0 User's Manual	2	8	-
Nero 6 Ultra Edition QuickStart Guide	7	3	-
TreePad Lite 2.9.5 User Guide	1	9	-
Total	**25**	**25**	**-**

Coverage is an interesting topic because its relationship with positioning seems very often to be inverse. When screen captures use positioning well (used in a procedure and they are an integral part of the instruction), the coverage is not used well to convey a good mental model, and vice versa. This presents the technical communicator with a dilemma: how to support both of these roles?

Below is an example of a screen capture that has good coverage that focuses on the relevant parts of the program and gives the user a good understanding of how to follow the procedure given in the document:

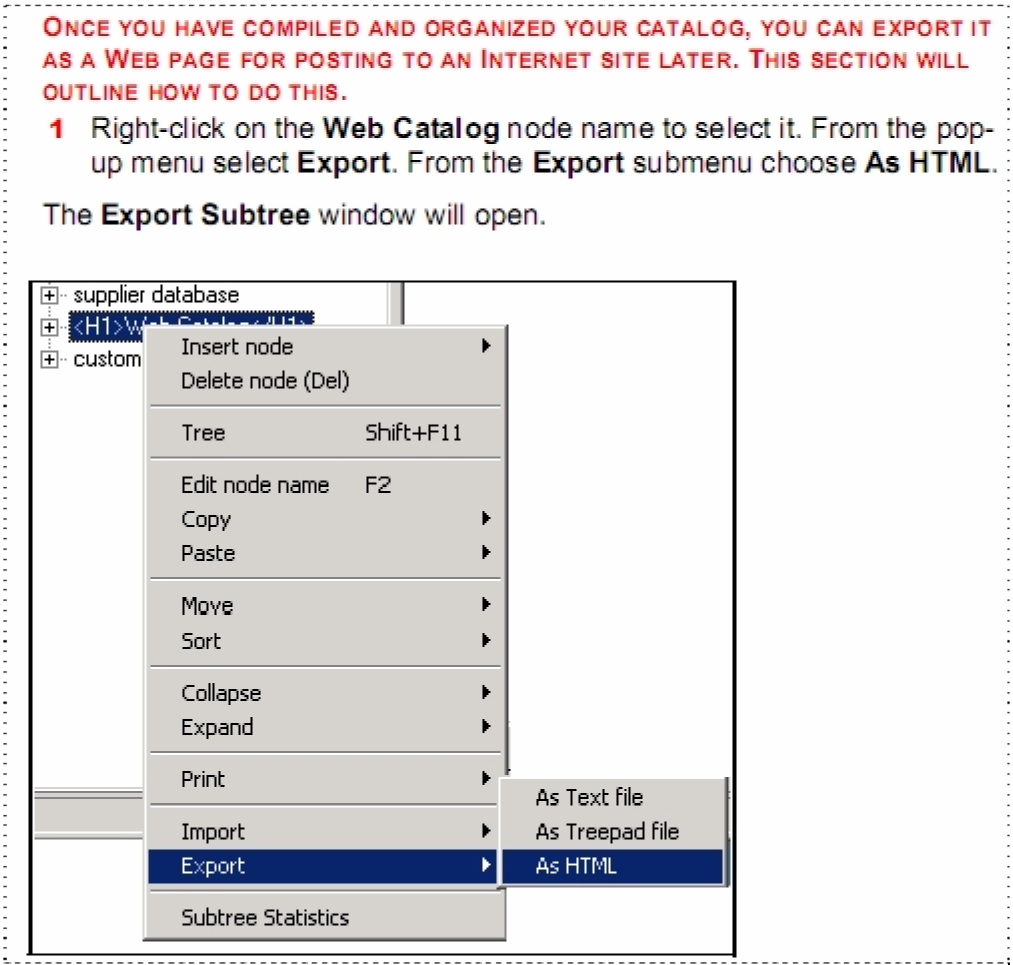


Figure 18 Coverage that helps the user to develop a mental model (from TreePad Lite 2.9.5 User Guide, screen capture 50 in the material)

The other extreme, often encountered in the material was the screen capture of a single icon or element, which does not help the user to develop a mental model of the program:


Spell-check all text in comments

1 Choose Edit > Check Spelling > In Comments, Fields, & Editable Text. If the PDF is open in a browser, make sure that the Edit toolbar is open, and click the Spell Check button.

Figure 19 A screen capture of a single icon - poor coverage (from Adobe Reader 8 User Guide, screen capture 10 in the material)

As for size, the size reduction in screen captures was easy to spot. Although this area, like many of the others, was better suited for analysing screen captures used in instructions, the results were quite clear. Out of 50 screen captures, the following number of screen captures were:

- using size in a way that supported the user: 23
- not using size in a way that supported the user: 7
- irrelevant: 20

In most cases size was taken into account, and there were even fewer cases where the size of the screen captures could have caused major problems for developing a mental model of the program. Most problematic screen captures were found in one document, Nero 6 Ultra Edition QuickStart Guide. The table below shows the breakdown of findings per document.

Table 7 Results for size per document

Name of the document	Uses size to support the user	Does not use size to support the user	Irrelevant
Adobe Reader 8 User Guide	7	-	3
Ad-Aware 2007 User Manual	-	1	9
Foxit Reader 2.0 User's Manual	8	-	2
Nero 6 Ultra Edition QuickStart Guide	2	5	3
TreePad Lite 2.9.5 User Guide	6	1	3
Total	**23**	**7**	**20**

Here is an example from the material where screen capture size could cause problems:

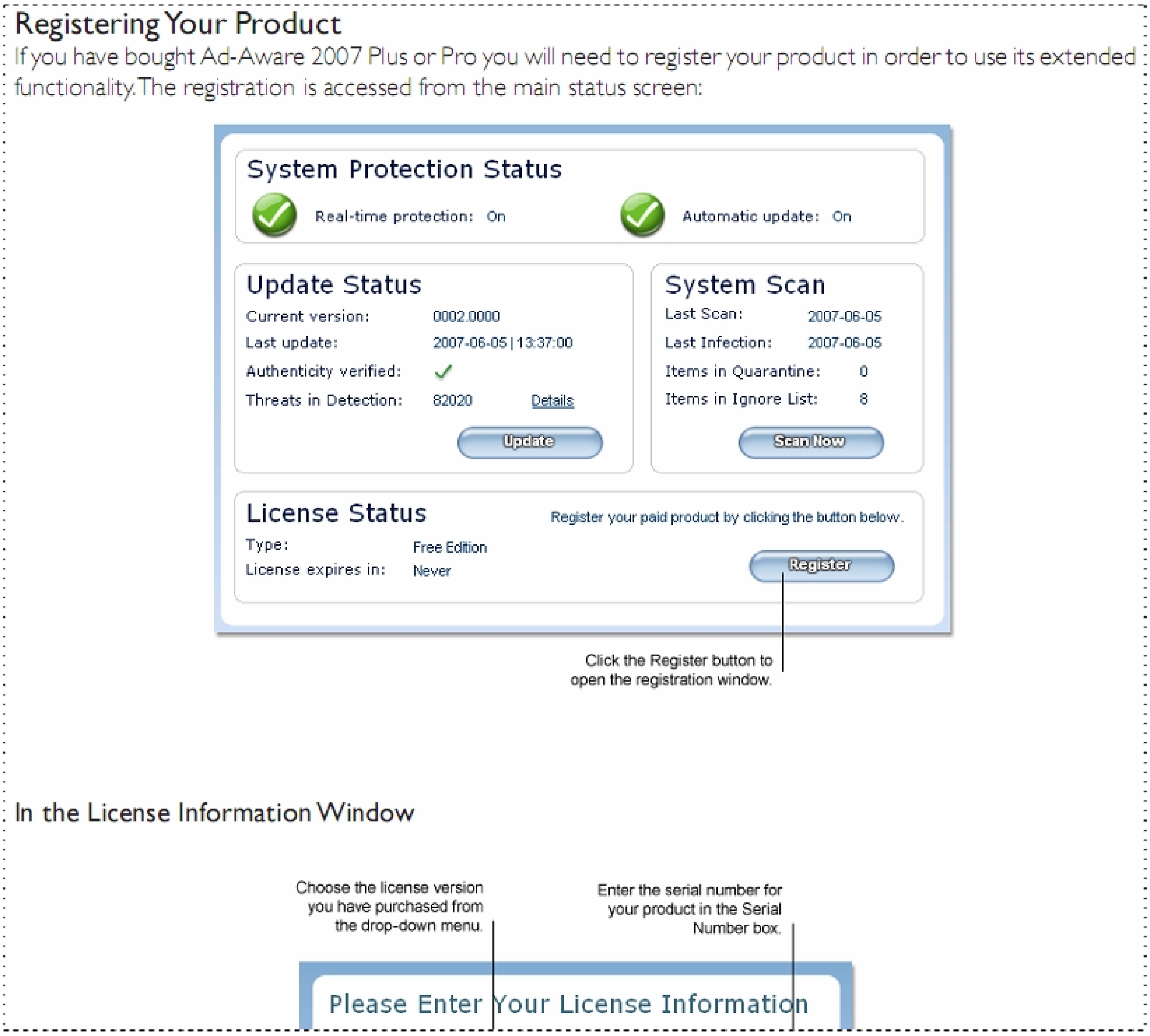


Figure 20 Example of a problem related to screen capture size (from Ad-Aware 2007 User Manual, screen capture 12 in the material)

Because the screen capture size is quite large in relation to the textual instructions, the second step in the procedure (titled 'In the Licence Information Window) is disconnected from the procedure. It is also problematic that there is too much white space between the procedures and the steps are not clearly numbered, but this was an aspect that was not analysed in this thesis. Even with proper numbering the screen capture size could have caused problems for developing a mental model.

One thing noted in the analysis was that many of the "irrelevant" screen captures were quite large, and could have caused problems in procedural information. These were labelled as "irrelevant" because they were not a part of an instruction, but were usually used in a descriptive role before the actual instructions. In this role they may actually have the correct size, as the user does not have any procedure to follow, and thus cannot lose the sense of connectedness that is needed in procedural information.

If we combine these results, it becomes evident that in the case material size was rarely a problem for developing a mental model of a program, but correct coverage was often missing, although there were differences between different documents.

5.3 Verifying Screen States (Analysis of Coverage)

Coverage, which was also analysed in the previous chapter, is also relevant for verifying screen states when screen captures are used to confirm that the user has followed a procedure correctly. This aspect of coverage was only relevant for screen captures used as a part of an instruction or procedure.

As expected, this analysis was fairly straightforward and also the results were quite clear. Out of 50 screen captures, the following number of screen captures were:

- using coverage in a way that supported the user: 3
- not using coverage in a way that supported the user: 27
- irrelevant: 20

In other words, there were only three cases where screen captures showed what would happen after a completed action. Here is one of the few examples where coverage was used in a way that would help the user to verify the screen state:

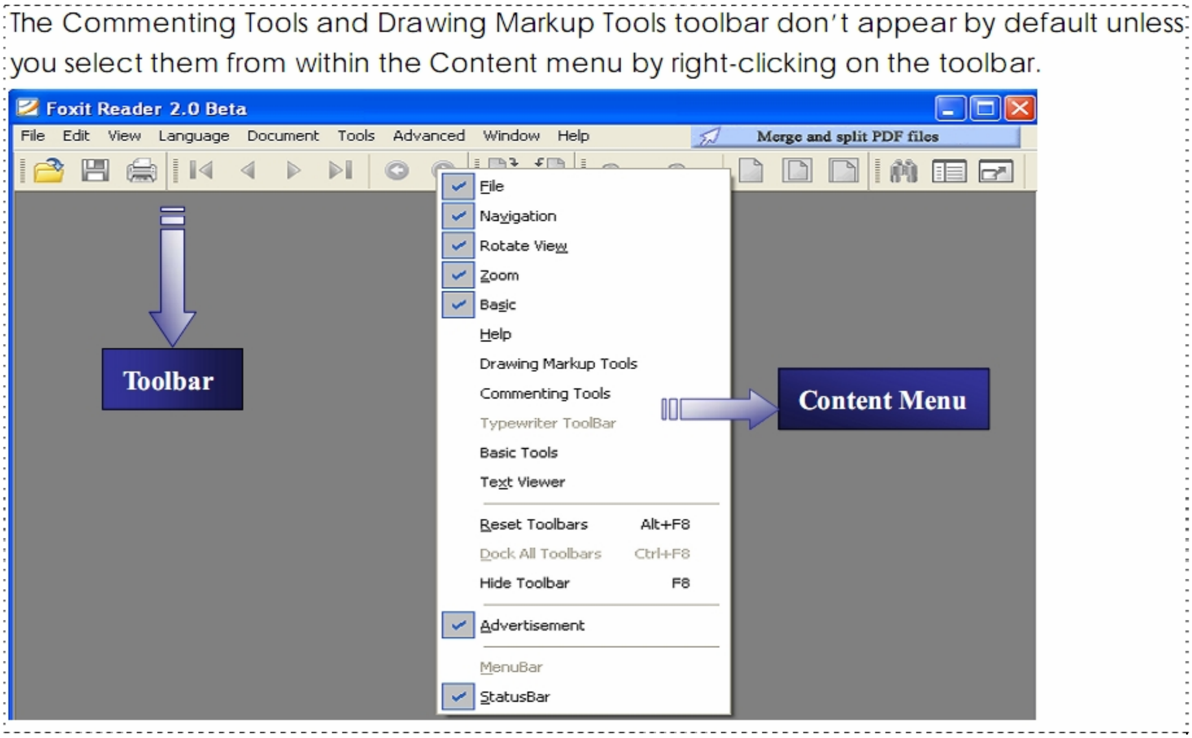


Figure 21 Using coverage to verify screen states (from Foxit Reader 2.0 User's Manual, screen capture 30 in the material)

Although this screen capture did not have a "before" screen capture, it shows the relevant items quite well. The problem here is that the coverage should be even smaller, so that only the relevant items would be shown.

Here is another example where the coverage is better because only part of the application is shown. This screen capture also shows both the "before" and the "after" state.

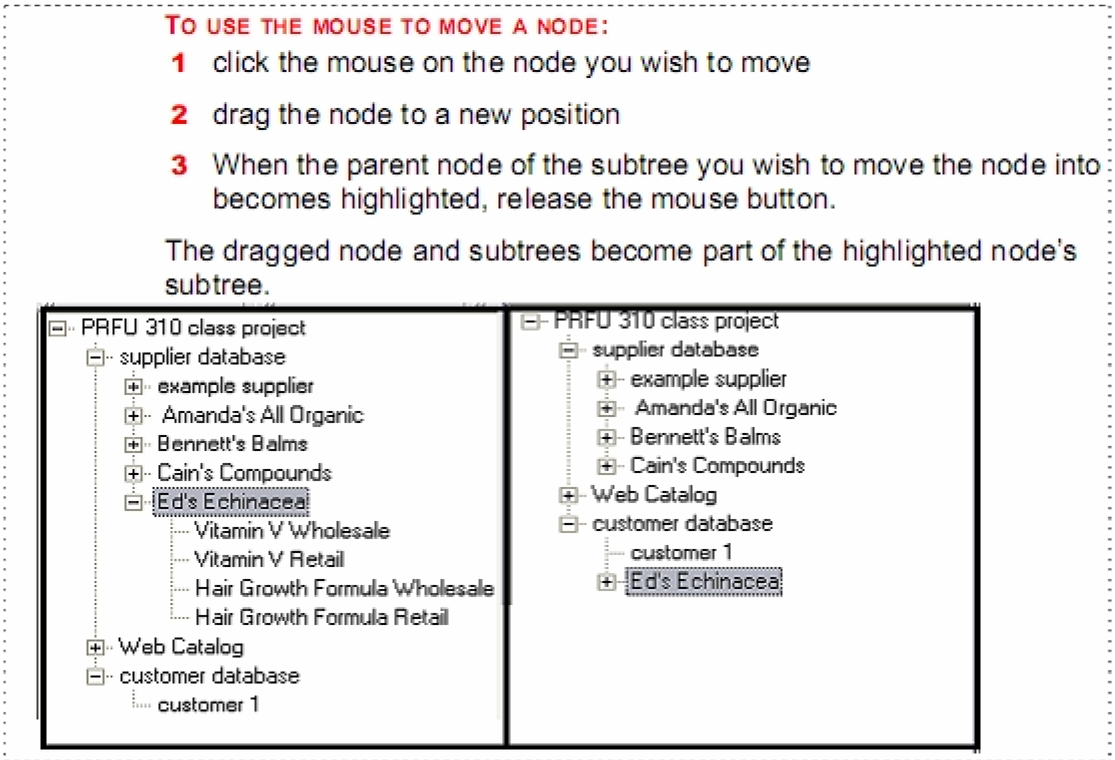


Figure 22 Better coverage (from TreePad Lite 2.9.5 User Guide, screen capture 48 in the material)

A breakdown of findings per document is not needed in this case because the findings are so clear.

5.4 Identifying and Locating Window Elements and Objects (Analysis of Cueing)

Some screen captures that used cueing were irrelevant for the analysis, but the reason differed from all the other areas that had irrelevant screen captures. There were a number of cases where cueing was irrelevant, most importantly in small screen captures that were an integral part of an instruction or small screen captures that showed a very small part of the program (one icon, pop-up window, pull-down menu). Small screen captures, such as these do not need cueing because they already show a very small part of the program. The analysis on whether the screen captures help the user to identify and locate window elements was done only for screen captures that could benefit from the use of cueing.

Here are the results in numbers. Out of 50 screen captures, the following number of screen captures were:

- using cueing in a way that supported the user: 20
- not using size in a way that supported the user: 21
- irrelevant: 9

The table below shows the breakdown of findings per document.

Table 8 Results for cueing per document

Name of the document	Uses cueing to support the user	Does not use cueing to support the user	Irrelevant
Adobe Reader 8 User Guide	3	7	-
Ad-Aware 2007 User Manual	7	3	-
Foxit Reader 2.0 User's Manual	4	-	6
Nero 6 Ultra Edition QuickStart Guide	3	5	2
TreePad Lite 2.9.5 User Guide	3	6	1
Total	**20**	**21**	**9**

The results for most individual documents were inconclusive, but it was evident that there were issues with consistency within individual documents. This was most apparent in Ad-Aware 2007 User Manual and Nero 6 Ultra Edition QuickStart Guide, where in many cases cueing was used effectively, but there were also cases where it was not used at all. The screen captures below are quite small, but it is possible to note that the upper screen capture on uses cueing, with hairlines identifying different parts of the screen. The lower screen capture does not have any added elements.

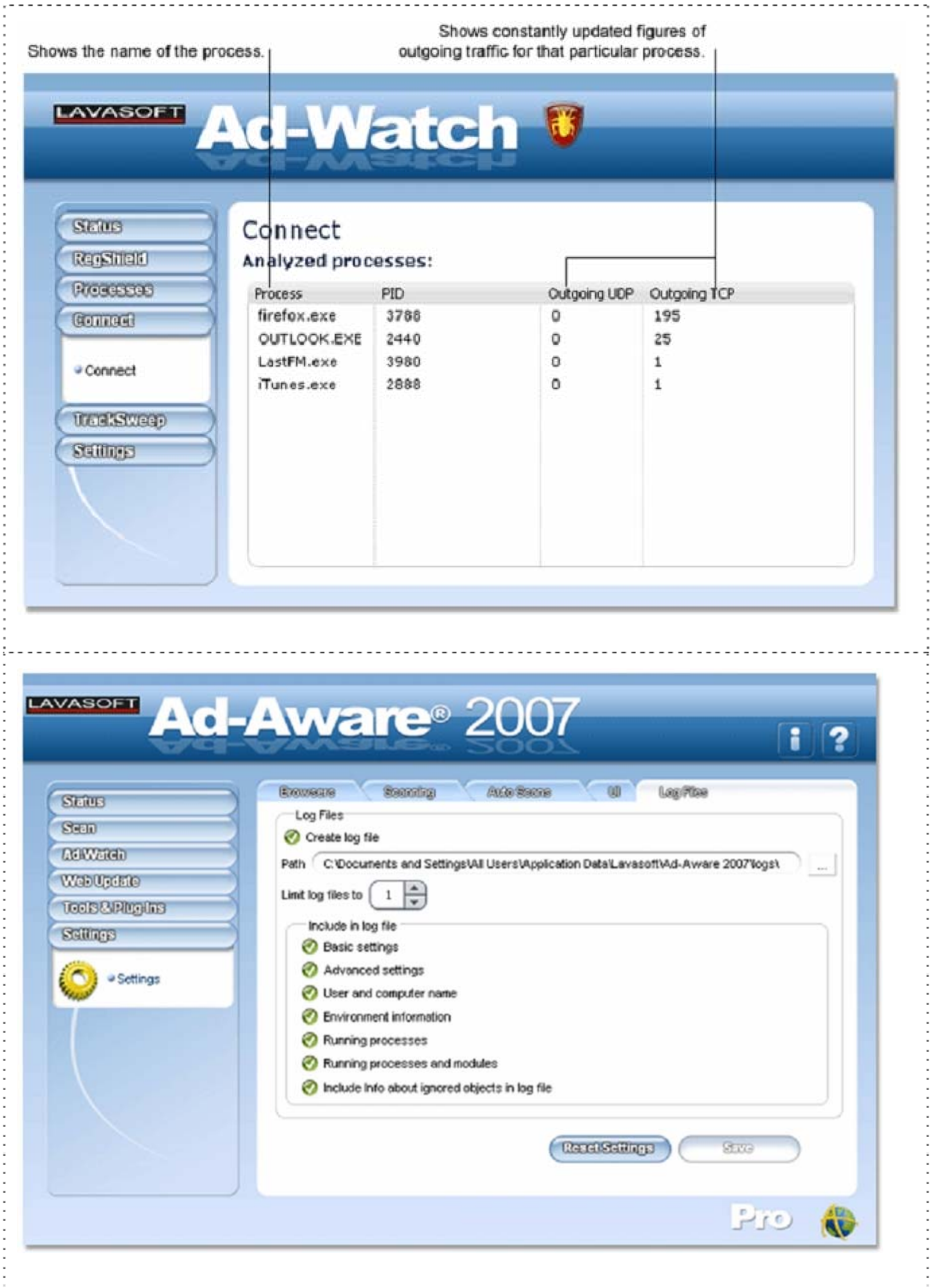


Figure 23 Inconsistent use of cueing (from Ad-Aware 2007 User Manual, screen captures 11 and 16 in the material)

Inconsistency was not the only problem with cueing. Active elements in the software can act as false cues, and there were some cases where false cueing could have hindered the user. The problem was not too widespread: there were four cases identified in the material. The following screen capture shows one example:

1. Start **Nero StartSmart**, select the 'Data' category and click on 'Make Data Disc'.

Figure 24 False cue (from Nero 6 Ultra Edition QuickStart Guide, screen capture 31 in the material)

Here you can see that the text instructs the user to click on ‘Make Data Disc’. The button for this action has not been identified in any way by using cueing. In addition to this, there is another problem in the screen capture because one of three icons at the bottom of the screen capture is active. You can see the rectangle around the icon. Active elements, similar to the one accidentally left in the screen capture above, can act as false cues. The user may not notice it, or it could prompt the user click in the wrong place. We cannot be certain what users would do in cases like these without actual usability testing in a laboratory, but it is quite possible that some users would click the falsely cued icon instead of the correct one, and might not even realise what prompted them to do so. This particular active element was easy to spot, but often they are more difficult to see. Usually there can be only one active item per window, but for example in the Windows operating system, it can still be hard to see whether the one of the buttons is active or not (see Figure 14 for example).

5.5 Findings and Carliner's Framework for Information Design

If we first place each of the design areas in Carliner's framework of information design, we should also be able to tell which areas of the framework are poorly supported and would need more emphasis from technical communicators. Note that some design areas and roles can be placed under one or more aspects of information design.

Physical

The results suggest that the area of physical information design could be better supported.

- Cueing clearly belongs under physical design because it helps the user to locate window elements and objects. The results were inconclusive, but there were problems in all except one document. This suggests that this area of physical design is not used consistently in most documents. This design area was the most problematic for physical design.

- Design areas that help the user to develop a mental model, namely coverage and size, can be placed here under physical design because a good mental model of a program will help the user to find information. Of these two areas, coverage was more problematic in 2 out of 5 documents that failed to support physical design. Size supported physical design in all except one of the documents studied.

Cognitive

The documents studied fared much better in the area of cognitive information design because the design areas that supported developing a mental model of a program had been taken into account fairly well.

- A good coverage helps the user understand information, because it helps to create a better mental model. In the analysis 2 out of 5 documents failed to support cognitive design, and there were some issues with consistent application of this design area.

- Size is the other area important for developing a mental model, but the findings suggest that this design area was used in a way that should not negatively affect the user's ability to understand information. There were few individual problems, but all except one document (Nero 6 Ultra Edition QuickStart Guide) passed this requirement for cognitive design.

Affective

Affective design was the most problematic of the information design areas.

- Coverage is also critical for verifying screen states. It helps the user by confirming that the user has acted correctly, motivating the user. In the analysis all the documents failed to support this aspect of affective design.

- Positioning belongs under affective design, because correct positioning gives visual cues to the user that help the user to switch attention. In this area the result was that 2 out of 5 documents clearly failed to use positioning to help motivate the user.

6 Discussion

In hindsight, it is evident that the focus of the analysis could have been on screen captures used specifically as part of procedural instructions where there are much less irrelevant findings. However, I argue that noting this problem is also a valid finding. Using screen captures only as descriptive elements limits the screen capture roles to developing a mental model of the program, and identifying and locating window elements and objects. Using screen captures as part of instructions should be more effective because all of the screen capture roles are then present.

In an analysis such as used in this thesis, it must also be kept in mind that the results from a limited set of material are not representative of the whole field of software documentation. Instead, the analysis proved that screen captures can be studied apart from users, and that the analysis can be repeated with a larger set of documentation, perhaps with more focus on select areas like procedures or instructions, or perhaps as comparison of different information design strategies. It was also possible to pinpoint problem areas in software documentation by studying screen captures. These common problems would have been difficult to spot by studying how users react to different screen captures. Analysing users is a very good method, but it would be more useful in a study that compares different solutions or information design strategies that technical communicators have chosen. The results of this thesis show which information design areas lack most support for the user, and what the individual problem cases within those areas are.

One of the stated aims of this thesis was to improve the quality of software documentation by identifying problems in the use of screen captures. It was also suggested that the solutions should be independent from tools or media used, and this was achieved. The following list is a set of guidelines based on the analysis, which professionals in the field might find useful. The overall result, when the findings were mapped to Carliner's framework, suggested that more emphasis is needed especially on affective information design, but also on physical information design. I present the conclusions in the order of importance in the three-part framework adopted from Carliner, and I have also added a general category for findings that affect all of the information design areas. After presenting the guidelines, I will discuss what aspects could benefit from a further study.

Affective (motivating the user to perform):

- Help users by verifying screen states by giving small "after" screen captures with correct coverage. This design area was neglected in all of the documents studied, and when it was used the coverage could have been better (see for example, Figure 21).
- Position the screen captures as integral parts of the instructions. There were clearly two different design strategies in the documents studied, but the design strategy where small screen captures are positioned in the instructions better motivates the user to switch attention (van der Meij and Gellevij, b: 531).

Physical (helping the user to find):

- Use cueing consistently. In the material there were documents that used cueing but a more consistent use was missing.
- Use active elements of the software product as visual cues, but be careful not to leave them in screen captures accidentally because they may distract and mislead the user (see for example Figure 24).

Cognitive (helping the user to understand):

- Focus on the relevant parts of program by giving enough coverage for the user. Positioning was often competing with this role. It was also noted in the material that many of the descriptive figures could benefit from a large size because there are no procedures to follow. A bigger descriptive screen capture shown before an instruction could help the user understand the program and develop a better mental model.
- Use coverage consistently. There were some issues with consistent use of coverage.

General design considerations that affect all the other areas:

- Use screen captures in instructions, not just as descriptive figures. This was clearly evident in the material as two of the screen capture roles become irrelevant if screen captures are just used as descriptive information.
- Presenting the same information in both a screen capture and in verbal form can help avoid splitting the user's attention. This is one of the possible ways to address the problems related to positioning and coverage that were noted in the material. This is further discussed below in relation to the level of user knowledge.

It is an interesting question if we should duplicate information, and whether we should use visuals or text. In 3.2.1, I presented a study conducted by van der Meij which found that users actually switched attention more effectively when using text-only instructions. Although it has not been explicitly concluded why this happens, this could also show that users' expectations are not necessarily aligned with what would be an efficient way of presenting information. The documentation that users expect to see might not be the best possible documentation for the user. Also, the different needs of expert vs. novice users need to be addressed. Different design strategies could help alleviate the problem that was noted for presenting information to users with different levels of competence. As was mentioned in 3.2.1, experienced users need less support for switching attention (fewer screen captures), whereas novice users benefit more from the use of screen captures. The dual coding theory mentioned in the Introduction suggests that users can simultaneously deal with textual and nonverbal objects. Technical communicators could take advantage of this by designing instructions that present both textual instructions and screen captures side by side, maybe in a two column layout. This could actually help deal with the dilemma that was noted in the analysis: the inverse relation between positioning and coverage. Positioning requires small screen captures that are used as an integral part of instruction, but coverage requires that you give the user large enough coverage that the user can develop a mental model of the program. Using textual instructions with small screen captures and larger complimentary screen captures could solve this particular problem, and the user would not need to split attention between text and figures that much because the information presented would be complimentary, not prerequisite for each other.

This would lead to an increase in the number of screen captures, which is another interesting area that needs more analysis - how much should we use screen captures? Horton tells us that with screen captures, less and fewer are more (146). His thoughts are based on observing test subjects in usability tests. He presents the following criticism for using screen captures:

- They spread apart the steps of a procedure. A half-page procedure now spans four pages. The user cannot as easily see how all the steps fit together. There is some evidence that this causes them to turn off their knowledge and common sense and follow the procedures blindly.

- Large "screen snapshots" can overwhelm the body text and even the headings. When this happens, the reader's eyes flit from picture to picture, sometimes ignoring critical text.

- "Screen snapshots" are used instead of other, more helpful graphics. Using lots of screen snapshots gives the writer false sense of security – they're meeting their minimum daily requirement for graphicality. […] (146)

I believe that these issues of cannot be solved by simply adding more screen captures, or removing screen captures. We need figures that fulfil the requirements of physical, cognitive, and affective information design. The problem with adding more complimentary information is information overload. According to Carliner, most information designers recommend solving this information overload by making changes in the physical design, structuring information, and omitting less essential information from documentation. Making changes in physical design would mean, for example, communicating through visuals rather than words (568). This would mean that in software documentation the amount of screen captures would grow in relation to the lines of text used. This would be a development that might not please Horton, for example.

I believe there is some balance that could be found by testing different information design strategies with users that have different levels of expertise. Instead of "dumb screen dumps" we will need smart screen captures. According to Carliner, a design theory should provide more than just series of guidelines, it should help the technical communicators choose the "right" solutions

(563). This stresses the importance of analyses such as the one conducted in this thesis, where the aim is both to map out different problem areas, and to present guidelines.

The guidelines above are based on the analysis of a fairly small set of documents, and it must be noted that Carliner's three-part model also presents many issues that were not studied, and could not be studied with the methods used in this thesis. He mentions, for example, cross-cultural communication, which I want to bring up here as a potential area that could be studied. This is also noted by Kress when he writes about the shift of dominance from textual information to visual information. Kress believes that this shift will have profound effects on "cultural and bodily engagement with the world" (1). Previous studies have indicated that users coming from different cultures expect different things of figures used in technical documentation (e.g., Fukuoka, Kojima, and Spyridakis). According to these studies, the main difference would seem to be the amount of figures expected, although the content can also differ. These observations do not, however, tell us whether the users' expectations correlate with what is "good" for users. This is also in parallel with Brasseur's ideas. The target of Brasseur's cultural critique is not the reader or the user, but rather the designer who has expectations of what the user needs, and how visual genres (e.g. charts, tables, diagrams, or figures) should be used.

These previous studies do not tell if cultural differences have a bearing on how figures or screen captures are perceived. Whether cultural aspects have a serious impact on the effectiveness of software manuals, is debatable. It would be my educated guess that users of software products are a much more homogenous group when compared to users of technical documentation in general. This could lessen the importance of cultural considerations. Most users of software products are also likely to be accustomed to how information is presented in software applications in general, and this could make cultural aspects less important for designing software documentation. I do not mean that cultural influence should be dismissed outright, but it might be of less importance in software documentation than, for example, the level of user knowledge (novice—expert).

Some technical communicators warn that technical communicators should not shift focus from "traditional documentation design" too fast (Gribbons and Elser 469). By traditional document design they mean documentation design that mainly concentrates on designing texts. Entry into this "new" field of visual design is a long process. To write better visual documentation we need to understand how documents affect human cognition on different levels. It is not enough to give practical advice to technical communicators on how to make screen captures or pictures more effective using specific tools. We need more understanding of pictures, text, and users of software documentation.

References

American National Standard for Telecommunication Glossary. Online. Institute for Telecommunication Sciences. Internet. 23 Oct. 2001. Available: http://www.its.bldrdoc.gov/projects/t1glossary2000/

Becker, M. W.., Pashler, H. "Volatile visual representations: Failing to detect changes in recently processed information." *Psychonomic Bulletin and Review* 9 (2002): 744-750. Available: http://www.pashler.com/Articles/Becker_Pashler2002.pdf

Brasseur, L. *Visualizing Technical Information – A Cultural Critique*. New York: Baywood Publishing Company Inc. 2003.

Carliner, S. "A Three-part Framework for Information Design" *Technical Communication* 47 (2000): 561-576.

Cook, G. "Principles for Research into Text Accessibility" in: Nyyssönen and Kuurre (eds.) 1995.

Fukuoka, W., Kojima, Y., and Spyridakis J. H. "Illustrations in User Manuals: Preference and Effectiveness with Japanese and American Readers." *Technical Communication* 46 (1999): 167-176.

Ganier, F. "Factors Affecting the Processing of Procedural Instructions: Implications for Document Design." IEEE Transactions on Professional Communication, vol 47, NO. 1 (2004): 15-26.

Gribbons, W. M. and Elser, A. G.. "Visualizing Information: An Overview of This Special Issue." *Technical Communication* 45 (1998): 467-472.

Hoft, N. L. *International Technical Communication*, *How to Export Information about High Technology*. New York: Wiley, 1995.

Hormby, T. "A history of Apple's Lisa, 1979-1986" Online. Available: http://lowendmac.com/orchard/05/1005.html.

Horton, W. "Visual Literacy, Going beyond words in technical communication. Dump the Dumb Screen Dumps." *Technical Communication* 40 (1993): 146-148.

Houghton-Alico, D. *Creating computer software user guides from manuals to menus.* New York: McGraw-Hill, 1985.

Kress, G. *Literacy in the New Media Age.* Routledge,an imprint of Taylor & Francis Books Ltd, 2003.

Lahti, M. *Efficiency of communication in software user documentation.* Tampere: A Pro Gradu thesis for the University of Tampere. 2000.

Mitchell, W. J. Thomas. *Picture Theory: Essays on Verbal and Visual Representation.* Chicago: University of Chicago Press, 1995.

McMurrey, D. A. *Power Tools for Technical Communication.* Heinle, 2001.

Oxford Dictionary of Computing. 4th ed. "Screen Dump." 1996.

Paivio, A. *Mental Representations*: *A Dual Coding Approach.* New York: Oxford University Press, 1986.

Pilto R. and T. Rapakko. "Testing Accessibility of Utility Texts – Work in Progress" in Nyyssönen and Kuure (eds.). 1995.

Price, J. How to Write a Computer Manual: A Handbook of Software Documentation. Menlo Park: Benjamin/Cummings. 1984.

Redish, J. **a**: "Understanding Readers." *Techniques for Technical Communicators.* 1993: 15-41.

Redish, J. **b**: “What is information design?” *Technical Communication* 47 (2000): 163-166.

Rossner, M. “Theories of Visual Rhetoric: Looking at the Human Genome.” *Journal of Technical Writing and Communication*. 31 (2001): 391-413.

Schriver, K A. *Dynamics in Document Design. Creating Texts for Readers*. New York: John Wiley & Sons, 1997.

Tufte, E. R. *The Visual Display of Quantitative Information*. Graphics Press USA, 2001.

American National Standard for Telecommunication Glossary. Online. Institute for Telecommunication Sciences. Internet. 23 Oct. 2001. Available: http://www.its.bldrdoc.gov/projects/t1glossary2000/

van der Meij, H. **a**: “Designs for screen captures.” Online. Available: http://www.stc.org/edu/47thConf/files/Screen%20Capture%20Designs.pdf

van der Meij, H. **b**: “Preventing or Overcoming Split-attention (and other user) problems.” Online. Available: http://www.stc.org/47thConf/postconf/User%20Problems.pdf

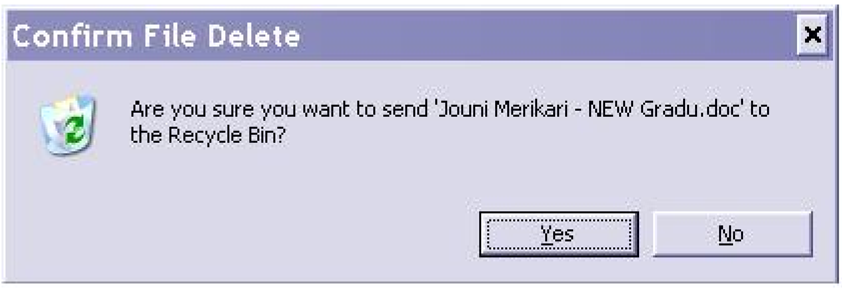
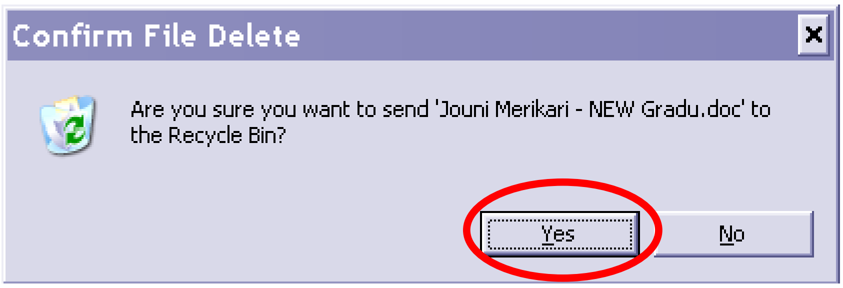
van der Meij, H., Gellevij, M. **a**: “Effects of Pictures, Age and Experience to Learning to Use a Computer Program.” *Technical Communication* 49 (2002): 330-340.

van der Meij, H. and Gellevij, M. **b**: “Screen Captures in Software Documentation.” *Technical Communication* 45 (1998): 529-543.

Vadén, T. *Digitaalinen Representaatio*. Tampere: Lecture material for the University of Tampere. 2000.

Glossary

In the following table, I list some of the terms defined in this document, or how they are used differently from a common definition. Some of these definitions are loaned from authors that have used similar terminology in their studies of screen captures; all such references are mentioned in the definition of the term.

Term	Definition
Active element	In screen captures: areas, buttons, fields or other elements of the program that are shown in an activated state in the screen capture. For example, in a typical dialogue box the user is presented with the choices OK and Cancel. Typically one of these is active and can be activated, for example, by pressing return. For example, here the button labelled 'Yes' is shown as active: 
Added elements	Elements that have been added to a screen capture to guide the user. These can be arrows, different colours, circles, lines etc. Following on the previous example, here I have added a circle identifying a specific part of the screen capture: 
Applications software	Programs that have an identified end use and are not directly related to the management of the computer system itself. (Houghton-Alico, 3)
Hardware	Physical elements of a computer, as opposed to computer programs (software).
Information design	A process of preparing communication products so that they achieve the performance objectives established for them (Carliner, 564). Divided into physical (helping users find information), cognitive (helping users understand information), and affective (motivating users).

Dialogue box	See, *window.*
Screen capture (screen dump, screenshot, screen snapshot)	A figure used mostly in software documentation. It is usually presents a snapshot view of one computer program, but may sometimes include parts of other programs or the operating system.
Software	Software is the data elements used for computer operation, as opposed to computer equipment, hardware. Software is often divided into two general categories: *systems software* and *applications software*. (Houghton-Alico, 1-2) Systems software means operating systems and the software "infrastructure" that enable applications (word processors, mail clients, games, etc.) to be run.
Software documentation	Software documentation can be divided into four main categories: design documents, user documents, project documents, and marketing material. (Lahti 7-8) In this thesis the term software document is used to refer to software user documentation only. Documentation that is not meant for end users, such as technical specifications, project documents, etc., is not specifically covered in this thesis.
Systems software	Tools necessary for managing applications software on a computer: language processors, operating systems, file management systems etc. (Houghton-Alico, 2)
Underutilisation	Underutilisation means that most users use only a fraction of the commands and functions available in a software product.
Window	Window is used in this thesis to refer to an area within the graphical user interface screen (or on the desktop) with which the user conducts a dialogue with a computer system. A window is the fundamental component of graphical user interfaces through which a user can view and manipulate an object. Generally, there are three types of windows: An *application window*, which is the focal point for user's activities A *document window*, which usually appears within the application window A *dialogue box*, which is a window with radio buttons, check boxes, text entry fields, and scrollable lists of choices for the user to choose from. (Macus A. et al, qtd. in van der Meij and Gellevij, b: 541)

APPENDIX – List of screen captures and results

Below is a list of all the screen captures analysed and the results for each design area. Results are summarised for each document and for the whole set. The page numbers mentioned here follow the conventions used in each of the case documents.

Table notation:

POS	Positioning
COV	Coverage
COV(M)	Coverage for developing a mental model of the program.
COV(V)	Coverage for verifying screen states.
SIZ	Size
CUE	Cueing
Y	The design area is used in a way that supports the user.
N	The design area is not used or used in a way that does not support the user.
N/A	Not applicable. The screen capture role is irrelevant in this case.

[A] The screen capture is not used as part of a procedure and thus the role of positioning is irrelevant in this case.

[B] The screen capture is not used as part of a procedure and thus the role of verifying screen states is irrelevant in this case.

[C] The screen capture is not used as part of a procedure and thus the role of developing a mental model is irrelevant in this case.

[D] The screen capture is a small screen capture that is either an integral part of an instruction, or would not benefit from cueing, and thus the role of identifying and location window elements is not relevant in this case.

Table 9 List of screen captures and findings

Figure	POS	COV		SIZ	CUE
		COV(M)	COV(V)		
Picture 1 - Adobe – page 23	Y	N	N	Y	N/A[D]
Picture 2 - Adobe – page 59	N/A[A]	N	N/A[B]	N/A[C]	Y
Picture 3 - Adobe – page 98	Y	N	N	Y	N/A[D]

		COV			
Picture 4 - Adobe – page 56	N/A[A]	Y	N/A[B]	N/A[C]	Y
Picture 5 - Adobe – page 82	Y	N	N	Y	N/A[D]
Picture 6 - Adobe – page 54	Y	N	N	Y	N/A[D]
Picture 7 - Adobe – page 117	Y	N	N	Y	N/A[D]
Picture 8 - Adobe – page 41	Y	N	N	Y	N/A[D]
Picture 9 - Adobe – page 61	N/A[A]	Y	N/A[B]	N/A[C]	Y
Picture 10 - Adobe – page 72	Y	N	N	Y	N/A[D]
Adobe **Results: Yes / No / Not applicable**	**7/-/3**	**8/2/-**	**-/7/3**	**7/-/3**	**3/7/-**
Picture 11 – Ad-Aware – page 28	N/A[A]	Y	N/A[B]	N/A[C]	Y
Picture 12 – Ad-Aware – page 3	N	Y	N	N	Y
Picture 13 – Ad-Aware – page 39	N/A[A]	N	N/A[B]	N/A[C]	N
Picture 14 – Ad-Aware – page 8	N/A[A]	Y	N/A[B]	N/A[C]	Y
Picture 15 – Ad-Aware – page 24	N/A[A]	Y	N/A[B]	N/A[C]	Y
Picture 16 – Ad-Aware – page 49	N/A[A]	N	N/A[B]	N/A[C]	N
Picture 17 – Ad-Aware – page 17	N/A[A]	Y	N/A[B]	N/A[C]	Y
Picture 18 – Ad-Aware – page 15	N/A[A]	Y	N/A[B]	N/A[C]	Y
Picture 19 – Ad-Aware – page 43	N/A[A]	N	N/A[B]	N/A[C]	N
Picture 20 – Ad-Aware – page 16	N/A[A]	Y	N/A[B]	N/A[C]	Y
Ad-Aware **Results: Yes / No / Not applicable**	**-/1/9**	**7/3/-**	**-/1/9**	**-/1/9**	**7/3/-**
Picture 21 – Foxit – page 30	Y	N	N	Y	N/A[D]
Picture 22 – Foxit – page 60	N	N	N	Y	Y
Picture 23 – Foxit – page 56	N/A[A]	N	N/A[B]	N/A[C]	Y
Picture 24 – Foxit – page 29	Y	N	N	Y	N/A[D]
Picture 25 – Foxit – page 70	Y	N	N	Y	N/A[D]
Picture 26 – Foxit – page 44	Y	N	N	Y	N/A[D]
Picture 27 – Foxit – page 48	Y	N	N	Y	N/A[D]
Picture 28 – Foxit – page 24	N/A[A]	Y	N/A[B]	N/A[C]	Y

		COV			
Picture 29 – Foxit – page 37	Y	N	N	Y	N/A[D]
Picture 30 – Foxit – page 57	N	Y	Y	Y	Y
Foxit Results: Yes / No / Not applicable	**6/2/2**	**2/8/-**	**1/7/2**	**8/-/2**	**4/-/6**
Picture 31 – Nero – page 21	N	Y	N	N	N
Picture 32 – Nero – page 38	N	Y	N	N	N
Picture 33 – Nero – page 11	N/A[A]	Y	N/A[B]	N/A[C]	N
Picture 34 – Nero – page 17	N/A[A]	N	N/A[B]	N/A[C]	N/A[D]
Picture 35 – Nero – page 23	N	Y	N	Y	Y
Picture 36 – Nero – page 18	N	N	N	Y	N
Picture 37 – Nero – page 43	N	Y	N	N	Y
Picture 38 – Nero – page 37	N	Y	N	N	Y
Picture 39 – Nero – page 12	N/A[A]	N	N/A[B]	N/A[C]	N/A[D]
Picture 40 – Nero – page 19	N	Y	N	N	N
Nero Results: Yes / No / Not applicable	**-/7/3**	**7/3/-**	**-/7/3**	**2/5/3**	**3/5/2**
Picture 41 - Treepad - page 16	N	N	N	Y	N
Picture 42 - Treepad - page 8-1	Y	N	Y	N	N
Picture 43 - Treepad - page 12-1	N/A[A]	N	N/A[B]	N/A[C]	N
Picture 44 - Treepad - page 8-2	N/A[A]	N	N/A[B]	N/A[C]	N/A[D]
Picture 45 - Treepad - page 4	N/A[A]	N	N/A[B]	N/A[C]	N
Picture 46 - Treepad - page 19	N	N	N	Y	N
Picture 47 - Treepad - page 12-2	N	N	N	Y	N
Picture 48 - Treepad - page 13	N	N	Y	Y	Y
Picture 49 - Treepad - page 15	N	N	N	Y	Y
Picture 50 - Treepad - page 20	N	Y	N	Y	Y
Treepad Results: Yes / No / Not applicable	**1/6/3**	**1/9/-**	**2/5/3**	**6/1/3**	**3/6/1**
Total: Yes / No / Not applicable	**14/16/20**	**25/25/-**	**3/27/20**	**23/7/20**	**20/21/9**

Suomenkielinen lyhennelmä

Johdanto

Kuvaruutukaappaukset ovat tietokoneohjelmista otettuja kuvia, joita käytetään ohjelmistodokumentaatiossa. Näitä usein käytettyjä kuvia tulisi tutkia lisää, koska niillä on rooleja, joita emme vielä täysin ymmärrä. Kuvia on toki aiemmin tutkittu teknisessä viestinnässä, ja esim. käännöstieteessä on tehty tutkimusta kuvan ja tekstin suhteesta, mutta itse kuvaruutukaappauksia on tutkittu suhteellisen vähän. Tämän tutkimuksen lähtökohta on nimenomaan teknisessä viestinnässä, joka on hyvin monitieteellinen, ihmis- ja käyttäjälähtöinen ala.

Meidän teknisten viestijöiden tulee osata suunnitella visuaalista informaatiota ohjelmistotuotteiden käyttäjille, mutta saamme harvoin käytännön tukea tähän työhön. Esim. Brasseur on tutkinut kulttuurillisia ja sosiaalisia rajoitteita, joita olemassa olevilla visuaalisen informaation esitystavoilla on, mutta Brasseur toteaa, että hänen tutkimuksensa ei anna vastauksia siihen miten tietoa lopulta tulisi esittää, koska tutkimuksessa keskitytään tutkimaan olosuhteita, jotka vaikuttavat tiedon esittämiseen (2). Visuaalisen informaation esittämiseen keskittyvä teokset (kts. esim. Tufte) eivät yleensä käsittele kuvaruutukaappauksia, eikä niitä yleensä ole suunniteltu nimenomaisesti teknisen viestinnän tarpeisiin. Kuitenkin Kress toteaa, että on käynnissä murros: olemme siirtymässä kirjoittamisen ja kirjan dominanssista kohti kuvan ja ruudun dominanssia (1).

Tutkimuksen tavoitteena on selvittää miten ohjelmistotuotteiden dokumentaatiossa käytetään tietokoneohjelmista otettuja kuvaruutukaappauksia ja kuinka hyvin käytetyt kuvaruutukaappaukset tukevat käyttäjiä. Tutkimuksen laajempana kehyksenä on Carlinerin kolmiosainen informaationsuunnittelun malli. Tarkemmin itse kuvaruutukaappauksia kuvaa van der Meij'n ja Gellevij'n teoria, joka kertoo mitä rooleja kuvaruutukaappauksilla on ja mitkä ovat näihin rooleihin vaikuttavat suunnittelun osa-alueet.

Kuvaruutukaappaukset ja ohjelmistodokumentaatio

Ohjelmistodokumentaatiolla tarkoitetaan tässä tutkimuksessa nimenomaan käyttäjille suunnattua dokumentaatiota ja lainaan tämän käsitteen määrittelyn Lahdelta (7-8). Ohjelmistodokumentaatiossa käytetään käyttäjän opastamiseen usein kuvaruutukaappauksia, jotka ovat todennäköisesti yleisin kuvatyyppi näissä dokumenteissa (Horton 146, Houghton-Alico ja Hoft 270). Kuvaruutukaappauksia käytetään myös määrällisesti paljon. Kun van der Meij ja Gellevij ottivat otoksen 100 ohjelmistodokumentista, he havaitsivat että 76 prosenttia sivuista sisälsi vähintään yhden kuvaruutukaappauksen (b: 529).

Miksi kuvaruutukaappauksia käytetään niin paljon? Hortonin mukaan ne tukevat käyttäjän kognitiota (146), ja tämä idea tulee myös esille van der Meij'n ja Gellevij'n teoriassa, jossa he esittelevät kuvaruutukaappausten neljä roolia:

- käyttäjän huomion siirtäminen
- mielikuvan luominen ohjelmasta
- muutosten havainnollistaminen ruudulla
- elementtien tunnistaminen ja paikallistaminen (b: 529)

Jokaisella näistä rooleista on yksi tai useita suunnittelun osa-alueita, jotka vaikuttavat siihen kuinka hyvin kyseinen rooli tukee käyttäjää. On mahdollista, että rooleja ja suunnittelun osa-alueita on muitakin, mutta tässä tutkimuksessa keskitytään nimenomaan tärkeimpiin näistä.

Kuvaruutukaappaukset ovat myös digitaalista informaatiota. Tämä on yksi tärkeimmistä syistä miksi tutkimuksessa ei oteta kantaa käytettyihin tiedonsiirtovälineisiin, työkaluihin tai prosesseihin, joilla kuvaruutukaappauksia tehdään. Digitaalinen informaatio ei ole sidoksissa näihin, koska mikä tahansa digitaalinen informaatio voidaan esittää millä tahansa digitaalisella järjestelmällä, esim. nuotit, bitit tai vaikkapa jono erivärisiä kiviä. Tämä tarkoittaa myös että digitaalista informaatiota voi käsitellä monenlaisilla työkaluilla, joten tiettyihin työkaluihin keskittyminen ei ole erityisen hyödyllistä. Järkevämpää on tutkia itse kuvaa, joka ei ole nyt (eikä tulevaisuudessakaan) riippuvainen tavasta jolla se säilytetään tai jolla sitä käsitellään.

Kuvaruutukaappausten roolit ja näihin liittyvät suunnittelun osa-alueet

Carlinerin kolmiosainen malli kuvaa informaationsuunnittelua. Se jakaa suunnittelun alueisiin joiden tarkoituksena on auttaa käyttäjää löytämään ja ymmärtämään tietoa (fyysinen ja kognitiivinen suunnittelu) sekä motivoida käyttäjää (affektiivinen suunnittelu) (564). Malli on joustava, joten sitä käytetään tässä tutkimuksena laajempana viitekehyksenä, johon materiaalista tehdyt havainnot voidaan luokitella.

Van der Meij'n ja Gellevij'n teoria kuvaruutukaappausten rooleista kuvaa parhaiten kuvaruutukaappauksia yksityiskohtaisella tasolla. Seuraavaksi esittelen lyhyesti jokaisen roolin ja siihen liittyvät suunnittelun osa-alueet.

Käyttäjän huomion siirtäminen on rooli, joka auttaa käyttäjää siirtämään huomionsa esim. kuvaruudusta manuaaliin ja takaisin. Tälle roolille kriittisin suunnittelun osa-alue on asettelu, jolla tarkoitetaan sitä mihin kuvaruutukaappaus asetetaan suhteessa ohjetekstiin. Kiinteästi ohjeen tekstin osaksi asetellun kuvaruutukaappauksen on havaittu helpottavan käyttäjiä siirtämään huomionsa kohteesta toiseen (b: 531)

Ohjelmistotuotteen käyttäjän tulee ymmärtää käytettävän ohjelman rakenne oppiakseen paremmin käyttämään sitä. Mielikuvan luominen ohjelmasta on toinen kuvaruutukaappausten rooleista ja muista rooleista poiketen sitä tukee kaksi suunnittelun osa-aluetta: kuvakoko ja kuvan kattavuus (b: 531).

Muutosten havainnollistaminen ruudulla on tärkeää, kun käyttäjän tämän tulee varmistaa toimineensa oikein. Käyttäjä kykenee paremmin havaitsemaan muutokset, kun kuvaruutukaappauksessa esitetään onnistunut lopputulos. Kuvan kattavuus on oleellista myös tälle roolille (b: 536).

Elementtien tunnistaminen ja paikallistaminen on rooli, jota tuetaan merkitsemällä käyttäjälle oleellisia elementtejä kuvaruutukaappauksessa (b: 537). Näin käyttäjän löytää oleellisen informaation nopeammin.

Kun roolit ja niihin liittyvät suunnittelun osa-alueet kootaan yhteen, saadaan aikaiseksi seuraava taulukko:

Rooli	Suunnittelun osa-alue
Huomion siirtäminen	→ Sijoittelu
Mielikuvan luominen ohjelmasta	→ Kuvan kattavuus, koko
Muutosten havainnollistaminen	→ Kuvan kattavuus
Elementtien tunnistaminen ja paikallistaminen	→ Merkitseminen

Materiaali ja metodit

Aineistoksi valitaan käyttäjille tarkoitettuja ohjelmistodokumentteja, jotka ovat vapaasti saatavilla Internetistä. Kriteerinä on myös se että kuvatut ohjelmistotuotteet eivät saa olla pitkälle erikoistuneita ammattiohjelmia. Valitut dokumentit ovat:

- Adobe Reader 8 User Guide, PDF-lukijan käyttöohje
- Ad-Aware 2007 User Manual, vakoiluohjelmien esto-ohjelman käyttöohje
- Foxit Reader 2.0 User's Manual, PDF-lukijan käyttöohje
- Nero 6 Ultr Edition QuickStart Guide, CD/DVD-poltto-ohjelman käyttöohje
- TreePad Lite 2.9.5 User's Guide, informaationhallintaohjelman käyttöohje

Tutkimuksessa otetaan tarkasteltavaksi kymmenen kuvaruutukaappauksen satunnaisotanta jokaisesta dokumentista. Näin on mahdollista analysoida myös pitkiä dokumentteja, joiden sisältämien kymmenien tai satojen kuvaruutukaappausten analysoiminen olisi muuten liian työlästä pro gradu -tasoisessa työssä.

Metodi tutkimukseen saadaan muokattua van der Meij'n ja Gellevij'n teoriassa esitellyistä suunnittelun osa-alueista (kts. taulukko yllä). Aineiston kuvat analysoidaan suhteessa jokaiseen osa-alueeseen ja tulokset esitettiin kvantitatiivisessa muodossa, sekä kokonaisuutena että dokumenteittain. Lopuksi on tarkoitus koota tulos yhteen ja katsoa miten se jakautuu suhteessa Carlinerin informaatiosuunnittelumalliin.

Tulokset

Tutkimuksen lähtökohtana oli selvittää miten hyvin kuvaruutukaappaukset toimivat rooleissa, jotka tukevat käyttäjää. Jos selkeitä ongelma-alueita löytyisi, niin tarkoitus oli myös esittää mihin pitäisi kiinnittää enemmän huomiota käytännön kirjoitus- ja suunnittelutyössä. Tutkimus osoitti, että vaikka dokumenttien välillä oli selkeitä eroja johtuen dokumenteissa valitusta tavasta esittää informaatiota, niin oli mahdollista havaita mitkä alueet parhaiten tai heikoiten tukivat käyttäjien tarpeita. Ongelmallista tarkastelun kannalta oli se, että useat valituista kuvaruutukaappauksista olivat ns. kuvaavaa informaatiota, eivätkä siis osa varsinaista käyttäjälle annettua toimintaohjetta. Tällöin useita kuvaruutukaappauksia ei voitu enää tarkastella käyttämällä van der Meij'n ja Gellevij'n teoriaa, koska useimmat teoriassa kuvatut suunnittelun osa-alueet ovat oleellisia ainoastaan käyttäjän seuratessa jotain tiettyä toimintosarjaa eli proseduuria. Näissä tapauksissa kuvaruutukaappaus merkittiin epäolennaiseksi ko. osa-alueen kannalta.

Käyttäjän huomion siirtämisen osalta ei ollut mahdollista sanoa tuettiinko tätä roolia tutkituissa dokumenteissa vai ei, mutta selkeitä dokumenttikohtaisia eroja oli mahdollista havaita. Kaksi dokumenteista selkeästi tuki tätä roolia ja kaksi ei tukenut.

Toinen rooleista, eli mielikuvan luominen ohjelmasta, oli hyvin tuettu kuvien koon osalta, mutta kuvien kattavuus oli usein riittämätön tukemaan käyttäjien kognitiota, vaikka yksittäisten dokumenttien välillä olikin eroja.

Muutosten havainnollistaminen oli heikosti tuettu kaikissa dokumenteissa. Koko aineistosta löytyi vain kolme tapausta, jossa käyttäjälle esitettiin kuvaruutukaappaus, joka tuki tätä roolia.

Elementtien tunnistamista ja paikallistamista oli tuettu kaikissa dokumenteissa, mutta useimmissa dokumenteissa oli myös kuvaruutukaappauksia, jossa tätä roolia ei tuettu. Tulosten perusteella ei ole mahdollista sanoa tuetaanko roolia hyvin ohjelmistodokumentaatiossa, mutta on selvää, että tätä roolia ei tuettu johdonmukaisesti.

Kun nämä tulokset jaettiin Carlinerin informaationsuunnittelumalliin, pystyttiin toteamaan, että heikoiten käyttäjää tuettiin niillä kuvaruutukaappausten suunnittelun osa-alueilla, jotka kuuluvat ns.

affektiiviseen suunnitteluun. Seuraavaksi eniten ongelmatapauksia oli fyysisen suunnittelun puolella, mutta kognitiivinen suunnittelu otti käyttäjät huomioon parhaiten.

Loppupäätelmät

Tutkimuksen lopputulos olisi voinut olla tarkempi, jos analyysin ulkopuolelle olisi rajattu deskriptiivisessä roolissa käytetyt kuvaruutukaappaukset. Pidän kuitenkin tätäkin havaintoa tärkeänä tuloksena, koska kuvaruutukaappausten käyttäminen vain osana kuvaavaa tietoa rajoittaa niiden tehokkuutta.

On muistettava, että tämä rajallisella aineistolla toteutettu tutkimus ei anna sellaisia tuloksia, joita voitaisiin yleistää koko ohjelmistodokumentaatioon. Tutkimustuloksista oli kuitenkin johdettavissa johtopäätöksiä siitä mihin informaationsuunnittelun osa-alueisiin pitäisi kiinnittää lisää huomiota käytännön kirjoitustyössä ja mitä yksittäisiä ongelmia tekninen viestijä voi kohdata. Esitän löydökset Carlinerin mallin mukaisesti niin, että ensimmäisenä on mainittu eniten parannuksia tarvitsevat osa-alueet.

Affektiivinen suunnittelu:

- Auta käyttäjiä havaitsemaan muutokset esittämällä toiminnon lopputulos kuvaruutukaappauksella, jossa kuva kattaa oleellisen lopputuloksen. Tämä oli heikosti tuettu aineistossa.
- Käytä kuvaruutukaappauksia ohjeen osina. Aineistossa havaittiin kahta ei lähestymistapaa, mutta aikaisempien tutkimusten perusteella kuvaruutukaappausten asettaminen osaksi toimintoa tukee paremmin käyttäjän toimintaa (van der Meija ja Gellevij, b: 531).

Fyysinen suunnittelu:

- Käytä merkitsemistä johdonmukaisesti. Tätä tehokeinoa käytettiin materiaalissa, mutta ei johdonmukaisella tavalla.

- Käytä tietokoneohjelmien aktiivisia elementtejä visuaalisina merkkeinä, mutta varo jättämästä näitä vahingossa kuvaruutukaappauksiin, koska harhaanjohtavat aktiiviset elementit voivat ohjata käyttäjän huomion väärään kohteeseen.

Kognitiivinen suunnittelu:

- Kuvaa sellainen osa ohjelmasta, joka kattaa tietokoneohjelman oleelliset osat.
- Käytä kattavuudeltaan oikeanlaisia kuvia johdonmukaisesti. Materiaalissa oli joitain ongelmia johdonmukaisuuden kanssa.

Yleisesti voidaan todeta, että kuvaruutukaappauksia tulisi käyttää osana ohjeita, ei ainoastaan kuvaavana informaationa. Saman informaation voi myös esittää rinnakkain sekä kuvana että tekstinä.

Löydösten pohjalta havaittiin myös potentiaalisia jatkotutkimuksen kohteita, joista tärkeimmät ovat tekstin ja kuvan suhteen tutkiminen, kulttuurin vaikutus, ja erilaisten suunnittelustrategioiden testaaminen käyttäjillä.