

A digital annotation system
for Apple Mac OS X.

NEON

► NAME:

Meikel Steiding

► SCHOOL:

**Thesis at Fachhochschule
Lübeck, Germany**






► PROFESSORS:

**Prof. Dr.-Ing., Dipl.-Ing. Hinrich Ecklundt
Herr Prof. Dr.-Ing. Hermann Hochhaus**

► DATE:

2005

CONTENT

004 	Introduction	What is Interaction Design?
008 	Research	The evolution of the window Form&Functions of annotations Reading mechanism on paper and online Relationship between paper and electronic documents From personal to public annotations Effect of annotations on students
040 	Analysis	Requirement list MATE PADD XLIBRIS Adobe Acrobat
071 	Results	Neon 1.0 Goal-Directed design Neon 2.0 Scenarios Requirement List
119 	Outro	Final words Appendix A: CD content Appendix B: The user test Appendix C: References

01

Why is it not possible to annotate any text on current computer systems? This thesis is about an interaction design concept. What does that mean?

INTRODUCTION

01 Since the presentation of the first Personal Computer scientists and engineers talked about the paperless office. They predicted that the new computers with their office&desktop paradigms will vanish simple paper. With the introduction of the internet those scientists and engineers went even further. They talked about digital libraries and hypertext applications. These new possibilities will change the concept of libraries, change the relationship of reader and author and some even said that in the near future, books will be obsolete.

But looking at todays document centric offices shows that none of it has really happened. The paper is all time present and book stores are still in the shopping malls. Obviously the computer revolution has not began yet and many people question if it ever will.

In 1984 Apple Computers Inc. introduced a computer named Macintosh. This computer introduced the Graphical User Interface to the mass market and made computer finally useable for everyone. Comparing the computers from that time with the computers today reveals that the speed and memory has been increased tremendously. New ways of transferring electronic information have been invented and the amount of computers in public and private environments multiplied.

What happened to the Graphical User Interface? Did the GUI evolve in equivalent speed? The basic interaction paradigm has not changed. The WIMP (Windows, Icons, Menus, Pointer) paradigm still exists. The interaction between these elements have been improved but the metaphors (i.e. trashcan) or idioms (i.e. double-click) are still main interface items today. Around 1993 color screens became affordable and interface designers added color coded feedback into the operating systems. With the introduction of Mac OS X, Apple Computers introduced native transparency into the man-machine interface. This transparency is widely used for decoration only. Volume settings and status information are fading in or windows get a dimension level through the use of drop shadows. Rare are examples where transparency is used to improve basic interaction with the computer system.

But what about the paperless office? Paper has still some incredible simple advantages and the actual computer systems are no real concurrence to that simplicity. Computers allow the user to create computer animated movies, music albums with virtual orchestration, simulating complex nature processes but highlighting words in a text or writing notes in the margin of a webpage is impossible. Simple annotations are still a main feature of paper. People print out their emails and hypertext documents to simply review the written text. Being able of annotating directly in the computer could have a great impact on the efficiency of working with paper and digital documents together. The requirements for a computer interface in 1984 were completely different then the requirements for a satisfying user experience today. It is time to question some of the common interface habits to adjust the current man-machine interface towards todays requirements. Annotations are only one example.

This thesis will show that annotations are part of the daily life in offices and universities. Building a mark up feature into the operating system might improve the work with paper and digital documents. Through the research in this thesis a requirement list for a digital annotation system will be build. This list will then be compared to the existing digital annotation systems and finally a mark up solution will be presented based on the build up requirement list.

WHAT IS INTERACTION DESIGN?

01.2 ▶ A digital annotation system would be a substantial addition to the current man machine interface. Since the topic of annotation is already widely known and used in the real world, transferring annotations to the digital world is a challenge from an interaction point of view. This thesis is about the Interaction Design of such a system. But what exactly does that mean?

‘Defining and designing the behavior of artifacts, environments, and systems, as well as the formal elements that communicate that behaviour’ is called Interaction Design^[1].

This discipline is mostly concerned with

- a definition of the products form in relation to its behavior and use.
- the anticipation on how the use of the product will affect human relationships and understanding
- the exploration of the dialogue between products, people and contexts

Interaction Designers are an interface to improve the complete design and development process. They are positioned between the user interface and the development team and help to anticipate expected and unexpected behaviors of a given product. Identifying early in the development how an application will behave tells the designer and developer how they need to design the system.

Good Interaction Design

- communicates a systems interactivity and functionality
- defines behavior that communicate a systems response to user interactions
- reveals both simple and complex workflows
- informs users about system state changes
- prevents user error

■ [1]

compare Alan Cooper, About Face 2.0, p.XXIX

The Interaction Designer is a translator in the human-machine communication. He/she needs to understand the properties and behaviors of both parties to ensure the effective communication between the machine and the user. This communication is described through actions and reactions. To overview the big picture he/she needs to be able to analyze how interaction unfolds over time and build a robust structure to let the user act in that environment errorless. One main part to reach these goals is to communicate the current state of the application. This allows the user to comfortably understand what type of operations are possible or appropriate at any given time. This comfort will end up in a workflow which the Interaction Designers task is to create. It is similar to the work of a film director who connects individual shots into scenes, and scenes into movies. The Interaction Designer uses individual screen elements to create screens and screens to workflows and workflows for fulfilling the users goal. But as with everything in life things can go wrong. Anticipating those errors and build solutions around it, is also part of the daily process and ensures that both the user and the system can easily recover.

Interaction designs the structure and behaviors of interactive products and services and users interact with those products and services.

Interface paradigms

A common misunderstanding is that good interface or interaction design is based on metaphors. To make complex processes more understandable it is quite common to borrow real world elements to build a comprehensive bridge for the user to let them guess the behavior and its task in advance. But metaphors lead to problems which are very hard to control. Alan Cooper stated out in the Visual Basic Programmers Journal in 1995: “*Searching for that guiding metaphor is like searching for the correct steam engine to power your airplane, or searching*

for a good dinosaur on which to ride to work.”^[1] Since this thesis is closely connected to something that exists in the real world but is missing in digital systems, the use of metaphors look promising and logical. But the advantages of metaphors are only of shortsight nature. A new user might benefit from the relation between the complex computer architecture and the already known item from the real world but after the new user explored the interface and understood how it works the developers and the users are tight to this real world cage.

There are actually three dominant interface paradigms in the conceptual and visual design of man machine interfaces: implementation-centric, metaphors and idioms.

Implementation-centric user interfaces are widespread in the current computer industry. These interfaces are based on the idea how things work. The user needs to know how they are built in order to use such interfaces. These software products have one button per function, one dialog per module of code and the commands and processes precisely echo the internal data structure. The user needs to understand the software in order to use it but “users would rather be successful than knowledgeable”^[2]. Implementation-centric interfaces are mostly the result of engineers building it. They are on the first hand easy to construct and since engineers like to know how things work the implementation-centric paradigm is very satisfying to them. But most users do not have the time nor the desire to explore the softwares inner-workings, they would be rather successful than knowledgeable a preference that is often hard to understand for engineers. This might be one reason why the implementation-centric paradigm is so popular.

Metaphoric interfaces are often considered intuitive. The behavior of an interface item will be related to a real world object so the user might be able to understand its function before even using it. There is no necessity in understanding the underlying data scheme to use it so it is an improvement towards the implementation-centric approach. But one major problem is the metaphors scalability. Imagining a computer system that solely relies on the desktop metaphor would result in the question if the

user needs to turn on the light during working times in the night? How does he/she switch between an art focused desktop for manipulating images to a spreadsheet-like desktop? Does the user need to empty the trash first before he/she can add new things to it? Concerning the digital annotation system it is important to know the limits of metaphors. A piece of paper can get lost, a pencil can break or might be out of ink. Papers are not easily to copy nor are they well scalable. In the following research and interaction concept it is important to focus on the advantages of digital systems as well as advantages from the real world. It easily happens that an interface will be bend to fit the chosen metaphor. That is the point where metaphors fail.

Idiomatic Interface are based on the way the users learn and use idioms. There is no technical knowledge (implementation paradigm) or intuition of functions (metaphor) necessary. Idiomatic expressions do not provoke associative connections the way metaphors do. The user understands an idiom simply by learning not because he/she needs to understand something or make subliminal connections in their minds. Most items of todays Graphical User Interfaces are based on idioms. Title Bars from windows, close buttons, hyperlinks, scroll-bars, drop down menus, all these are examples for interface idioms. By looking at a scroll-bar nobody will realize how this element is going to behave but using it only once explains the whole item and users will keep that in mind easily. Nesting of folders or resizing a window is not a metaphor, it is an idiom. “All idioms must be learned; good idioms need to be learned only once.”^[3]

Most of what the user knows did he/she learn without understanding: things like faces, social interactions, attitudes, melodies, brand names, the arrangement of rooms and furniture in our houses. The user do not understand why someones face is composed the way it is but he/she knows it. That is the power idioms are using.

Before a digital annotation system can be designed it is important to understand the behavior and structure of annotations offline. Research is an essential part in the interaction design to fully understand the fundament the system is based on. For this thesis the research starts out with studies from professionals around the world looking on how annotations are done on paper and in what context people use them.

■ [1]

Alan Cooper, 'The Myth of Metaphor', Visual Basic Programmer's Journal, June 1995

■ [2]

Alan Cooper, About Face 2.0, p.248

■ [3]

Alan Cooper, About Face 2.0, p.251

02

A summary about the evolution of the most popular interface item.

THE EVOLUTION OF THE WINDOW

02 In the Online Encyclopedia Britannica a 'graphical user interface' is defined by:

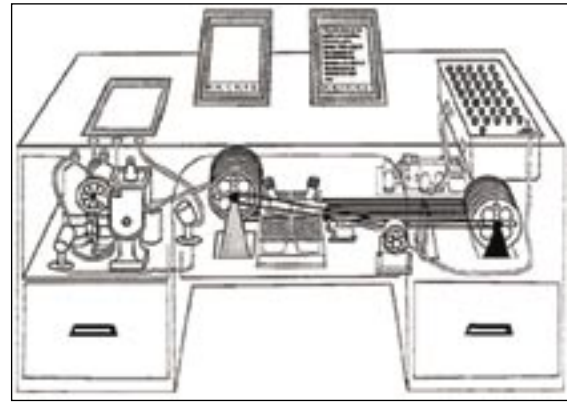
"A computer program that enables a person to communicate with a computer through the use of symbols, visual metaphors, and pointing devices."^[1]

Part of this GUI is an element which is commonly known as the window. This item is mainly used to provide space for displaying content or controls on a computer screen. In 1945 Vannevar Bush, at that time Director of the Office of Scientific Research and Development, envisioned about a system he calls 'memex'.

"[...] in which an individual stores all his books, records, and communications [...] It consists of a desk [...] On the top are slanting translucent screens, on which material can be projected for convenient reading. There is a keyboard, and sets of buttons and levers [...] if the user inserted five thousand pages of material a day it would take him hundreds of years to fill the repository [...] If the user wishes to consult a certain book, he taps its code on the keyboard, and the title page of the book promptly appears before him, projected onto one of his viewing positions [...] [with] one of the levers to the right he runs through the book before him, each page in turn being projected at a speed which just allows a recognizing glance at each. If he deflects it further to the right, he steps through the book ten pages at a time; still further at one hundred pages at a time. Deflection to the left gives him the same control backwards [...] he can leave one item in position while he calls up another"^[2]. This 'memex' vision is widely recognized as the first major mention of a personal computer.

Vannevar Bush served as a science advisor to both Franklin Roosevelt and Harry Truman and his greatest contribution were the oversight of the Manhattan Project (first atomic bomb) and second to build a structure for the science and technology efforts after World War II. A main part in this structure played the Advanced Research Projects Agency (ARPA).

In 1962 the ARPA turned its business into the computer science by establishing the Information Processing Techniques Office (IPTO). Before that time ARPA worked mostly on military projects like the M-16 rifle or



■ FIGURE 2.1 Memex

ballistic missile defense and tactical antitank weapons technologies. J.C.R. Licklider was the first director of this newly founded office. In 1959 Licklider wrote a book titled: 'Libraries of the Future'. This book was dedicated to Bush's 'Memex' concept and was about a new kind of library to the world's new knowledge system. Licklider got famous at ARPA for two basic concepts he evolved there. He had *"the first concrete proposal for establishing a geographically distributed network of computers."*^[3] This intergalactic network would first be seen as a form of computer time-sharing links and later be transformed into the ARPANET (later known as the Internet). His second concept was focused on the 'man-computer symbiosis'. He wanted a system that contribute, rather than replace, human cognitive processes. It should support real-time processing and interactivity and redefine the existing user input in the machine to user interface—the display—through graphical input and automatic speech recognition. Licklider wanted to realize *"interaction with a computer in the same way that you think with a colleague whose competence supplements your own."*^[4] All his ideas were matters of Human Computer Interaction (HCI) and so he defined the agenda of HCI for years not just for ARPA but even for all universities connected to the science structure established by Vannevar Bush.

In 1964 Licklider left the IPTO and announced Ivan Sutherland to its new director. Sutherland was known for

■ [1]

Encyclopedia Britannica Online, Graphical User Interface, 3.9.2005

■ [2]

Glenn R. Fong, ARPA Does Windows: The Defense Underpinning of the PC Revolution, p. 223

■ [3]

Glenn R. Fong, ARPA Does Windows: The Defense Underpinning of the PC Revolution, p. 225

■ [4]

Glenn R. Fong, ARPA Does Windows: The Defense Underpinning of the PC Revolution, p. 226

his 1962 Ph.D. project at MIT called Sketchpad. It was “one of the most significant developments in human-computer communication”^[1] yet and he was motivated to put HCI on a new level. First Sketchpad was the first to be seen computer with a monitor and direct feedback of the users work on the screen. Before that time people needed to wait for printouts in order to see their results. And second, Sketchpad was the first computer with a pointing device for user input. It was a light pen which allowed you to draw directly on the monitor. For the first time in the computer history the user saw the results of his/her movements directly on his/her screen instead of typing coordinates to express simple geometry. The manipulation was not limited to simple drawings also interaction was possible in the way that the user could grab things and drag them or rotate, expand and contract images.

But that was not enough Sketchpad was also the first computer with a rudimentary windowing system. Sketchpad allowed the user to split the screen into two different work areas which allowed comparing or even drawing on two ‘documents’ at the same time. Sketchpad was financed by the Navy, Army and Airforce.



■ FIGURE 2.2 Sketchpad

■ [1]

Glenn R. Fong, ARPA Does Windows: The Defense Underpinning of the PC Revolution, p. 226

■ [2]

Glenn R. Fong, ARPA Does Windows: The Defense Underpinning of the PC Revolution, p. 227

But Licklider did not only hire Sutherland he also hired a man called Robert Taylor and announced him as the associative director of IPTO. When Sutherland left the ARPA in 1966 Taylor became the director of IPTO. Taylor was also very inspired by Bushs’ ‘Memex’ thesis and the work of Licklider as well. In 1968 Licklider and Taylor published a document in which they describe a computer with a large television monitor, a keyboard and a “electronic pointer controllers called ‘mice’ [that could] control the movements of a tracking pointer on the TV screen.”^[2] And in 1971, while working at Xerox PARC already, he announced that he is going to build the computer described 1968 with the help of his colleagues at ARPA. The name for this machine will be Alto.

At that time ARPA was sponsoring a lot of universities who had HCI Programs running. One of those programmes was the LOGO project at MIT. In 1968 a person named Alan Kay visited the MIT and Seymour Papert showed him how kids were working with the computer used Logo:

“In 1968 I saw two or three things that sort of changed my whole notion of computing.[...] In 1968 I saw Seymour Papert’s first work with kids and LOGO, and I saw the first really great handwriting-character-recognition system at Rand. It’s a fabulous system. And that had a huge influence on me because it had an intimate feel. When I combined that with the idea that kids had to use it, the concept of a computer because something much more like a supermedium. Something more like a superpaper.”^[3]

After this impressive presentation Kay wrote a thesis about graphical object-orientation and was being awarded with a Ph.D. at the University of Utah. Right after this thesis which had an enormous amount of inspiration for further GUI development Kay worked on a new kind of programming language called Smalltalk. “Smalltalk is the first object-oriented programming language with an integrated user interface, overlapping windows, integrated documents, and cut&paste editor.”^[4]

In the July of 1970 Xerox decided to build a new research center based in Palo Alto. They hired Bob Taylor to start the ‘Computer Science Laboratory’. Taylor was

■ [3]

Scott Gasch, Alan Kay, Fall 1996

■ [4]

Xerox PARC–Palo Alto Research Center, About PARC, 12.8.2003

not sure about working for a company and asked Alan Kay if he would consult him. Alan agreed and started working on new projects aiming to bring the computer closer to kids. In 1972 Xerox PARC released Smalltalk for the first time. It was the beginning of a number of invention which were about to change the whole computer industry as seen at that time.

In the same year Xerox announced to work on a new kind of computer called the Alto. It was centred on the strategy of what they called 'distributed interactive computing'.



■ FIGURE 2.3 Xerox Alto

The Alto was 'distributed' in the way that the concept was bringing the computer itself from the basement onto the desktop. And it was 'interactive' on the first hand that is was 'networking' with other Altos via Ethernet (invention of Xerox PARC as well) and secondly that its graphical interface was responding directly and was user-friendly designed using the WIMP (Windows, Icons, Menus, Pointer) paradigm. The Alto was the first machine with overlapping windows. Before that time computers like Sketchpad had the ability to split the screen in several workspaces. These workspaces were kind of windows with the ability to be moved around the screen. With the introduction of the Xerox Alto those workspaces could overlap and had a menu bar and closing button. So all in all they were much more in the shape which is known as the window today.

The Alto had a full set of applications for word processing, graphics (including animation), printing, email,

and playing music. 2000 of those machines were built and used by universities, government and industry. The Alto was not commercially successful it was marketed for 'knowledge workers' but it was way too innovative for most people and with a price of over \$16,000 not really affordable. In 1981 Xerox introduced a renamed version of the Alto – the Xerox Star. This machine was marketed as "a new personal computer designed for offices [...] intended for business professionals who create, analyze, and distribute information."^[1] The Star was a commercial failure as well.

After the Apple II the CEO of Apple Computers Steve Jobs was concerned of losing market share if they would not come up with something completely new. The Lisa was already in development but was based on a command line interface like any other commercial machine at that time. In the beginning of the seventies Jef Raskin who had close contact to Xerox PARC told Steve Jobs to take a look at the Xerox Alto.

Steve Jobs: "But I heard a few times, people would tell me, hey there was these guys over in this garage at Xerox Parc you ought to go see them and I finally went and saw them and I saw what they were doing and it was better than what we were doing."^[2] After Apple arranged to give Xerox \$1 million of Apple Shares Xerox prepared a little presentation for Steve Jobs at PARC.

Steve Jobs: "And they showed me really three things. But I was so blinded by the first one I didn't even really see the other two. One of the things they showed me was object orienting programming they showed me that but I didn't even see that. The other one they showed me was a networked computer system...they had over a hundred Alto computers all networked using email etc., I didn't even see that. I was so blinded by the first thing they showed me which was the graphical user interface. I thought it was the best thing I'd ever seen in my life. Now remember it was very flawed, what we saw was incomplete, they'd done a bunch of things wrong. But we didn't know that at the time but still though they had the germ of the idea was there and they'd done it very well and within you know ten minutes it was obvi-

■ [1]

Glenn R. Fong, ARPA Does Windows: The Defense Underpinning of the PC Revolution, p. 217

■ [2]

Robert X. Cringely, Triumph of the nerds: THE TELEVISION PROGRAM TRANSCRIPTS: PART III

ous to me that all computers would work like this some day."^[1]

Xerox hold two meetings. One for Steve Jobs and one for some Apple executives and programmers. Right after those meetings where Apple saw Xerox first attempts of a Graphical User Interface (GUI) they rebuild the whole Lisa Project and made it the first Apple Computer with a GUI. But the Lisa hit the same destiny like the Xerox Star both introduced in 1983 – they were too expensive. They could not compete against the cheap IBM Computers with a command line interface. Obviously people were not willing to pay \$10.000 more to have an easier to use man-machine interface. That should change with the introduction of the Macintosh. At its release on January 24th 1984 the original Macintosh was priced at \$2,495. Xerox gave Apple never any blueprints or pieces of code. The whole Macintosh interface was build out of Apple engineers minds and Jef Raskin and Bruce Horn improved the Xerox significantly

In 1996 Bruce Horn said: *"Smalltalk has no Finder, and no need for one, really. Drag-and-drop file manipulation came from the Mac group, along with many other unique concepts: resources and dual-fork files for storing layout and international information apart from code; definition procedures; drag-and-drop system extension and configuration; types and creators for files; direct manipulation editing of document, disk, and application names; redundant typed data for the clipboard; multiple views of the file system; desk accessories; and control panels, among others. The Lisa group invented some fundamental concepts as well: pull down menus, the imaging and windowing models based on QuickDraw, the clipboard, and cleanly internationalizable software. Smalltalk had a three-button mouse and pop-up menus, in contrast to the Mac's menu bar and one-button mouse. Smalltalk didn't even have self-repairing windows - you had to click in them to get them to repaint, and programs couldn't draw into partially obscured windows. Bill Atkinson did not know this, so he invented regions as the basis of QuickDraw and the Window Manager so that he could quickly draw in covered windows and repaint portions of windows brought to the front."*^[2]

■ [1]

Robert X. Cringely, Triumph of the nerds: THE TELEVISION PROGRAM TRANSCRIPTS: PART III

■ [2]

Bruce Horn, On Xerox, Apple and Progress, 1996

With the Alto Xerox introduced overlapping windows. But in the Alto there was no refresh of windows in the background. The user needed to click inside one window to actually get the right content. Bill Atkinson on the Macintosh solved this problem by using 'Regions'. Overlapping windows can form quite complex shapes and since the Macintosh used rounded window corners the takes of a fast and subtle window refresh was not easy.

"The region data structure was a variable-sized list of what Bill called „inversion points“, the coordinate values where black changed to white or vice versa. Since most regions were mostly rectangular, there weren't many inversion points, so regions were quite compact. But occasionally, there were lots of inversions, like in a circle, so regions grew as necessary."^[3]

The Apple Macintosh was the first affordable Personal Computer which worked with windows as known today. The first version of Mac OS did not allow to maximize a window to fill out the whole screen nor did it allow to collapse a window to its title bar. These features were introduced in 1986 (maximizing windows) and 1995 (collapsing windows).

In 1995 with the release of Windows95 Microsoft presented the popular taskbar for the first time. Next to the possibility of closing or maximizing a window, the user could minimize the window into the taskbar. This taskbar lists all application and document windows and was Microsofts solution of handling the increasing amount of windows inside the GUI.

Summarizing the evolution of the window reveals that there has never been a single company/person which presented this basic interface item as an invention of their/its own. Windows are not a result from complex studies or user research. They simply grew out of the idea to have the possibility of splitting a screen in separate workspaces. Moving these workspaces around, overlapping them or even change their size were just logical steps developed by different software vendors.

These workspaces are what is called a window today. It's origin is a simple rectangle, filled with information—the so called content. There are different kinds of windows to draw the desired information in wether it is text, graphics, moving images or interaction items like buttons for example. The window is thus the equivalent to paper

■ [3]

Andy Hertzfeld, I still remember Regions, April 1982

in the real world. This analogy is clearly communicated in the design of the Graphical User Interface (GUI). The term documents is used for files which are opened in windows. The default icon for such documents is inspired by an empty sheet of paper.



■ FIGURE 2.4 Document Icons (Mac OS 9 / Mac OS X / Windows)

By looking at the windows today, what has changed throughout the time? First the user can open and close windows. Second he/she can maximizing them and third the user can minimize windows. In 1995 the user got the ability to collapse a window on a Mac OS System. It is similar of rolling in 90% of a sheet of paper while leave the top flat to still recognize the document. That proofed to be a quite efficient and popular way to control the increasing amount of windows on the desktops in Mac OS. Microsoft uses in comparison to collapsing windows their taskbar—introduced in 1995. This taskbar has more features then listing the minimized windows but its basic intention is the same—getting control of the increasing amount of windows on the computer.

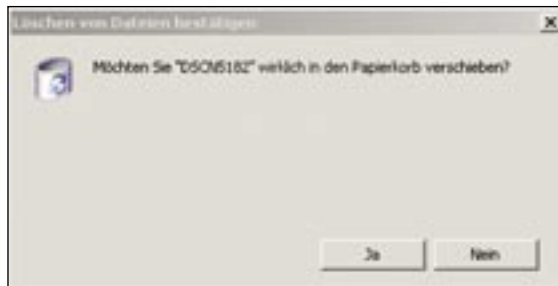
It seems that the windows' evolution stopped with the introduction of the Macintosh in 1984. Paper and windows are very different even though they serve similar purposes—providing information. The user can not rumple a window, nor can he/she discern or fold it. He/she can not throw it on somebody nor can he/she simply flip a window. All this is natural with a paper. The user even can not just write some stuff in the window independently of what it displays. This thesis is going to analyze the problem of annotating the content of windows. Every window should provide the user with the ability to annotate its content. Annotations are all time present on paper and a computer window should provide a similar functionality.

REQUIREMENT 001

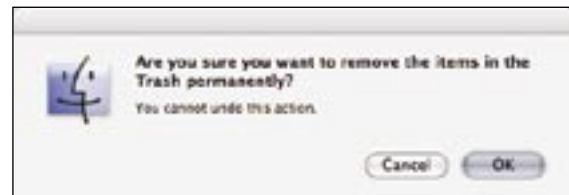
All time present possibility to annotate content in a computer window. It is part of its appearance.

WINDOW TYPES

Dialog Window:



■ FIGURE 2.5 Microsoft Windows Dialog Box



■ FIGURE 2.6 Apple Mac OS X Dialog Box

Palette Window:

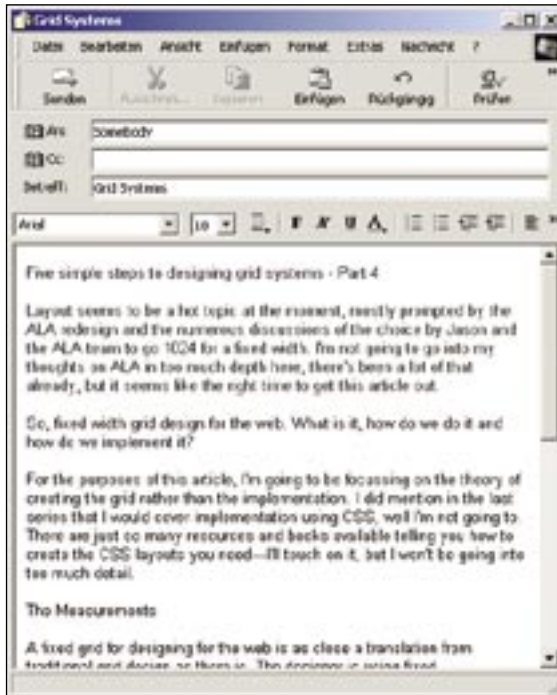


■ FIGURE 2.7 Microsoft Windows Preference Window



■ FIGURE 2.8 Apple Mac OS X Preference Window

Document Window:



■ FIGURE 2.9 Microsoft Windows Document Window

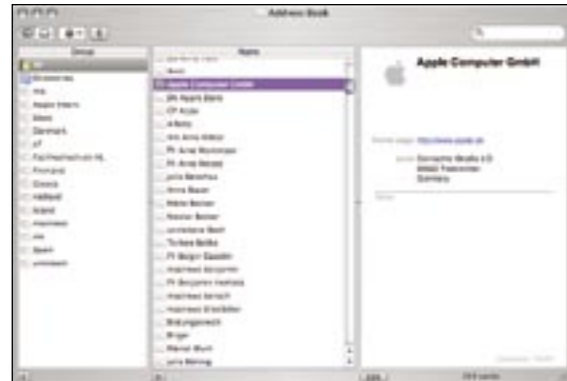


■ FIGURE 2.10 Apple Mac OS X Document Window

Application Window:



■ FIGURE 2.11 Microsoft Windows Application Window



■ FIGURE 2.12 Apple Mac OS X Application Window

03

Different forms serves different functions.

An analyzes based on research from Xerox PARC.

FORM & FUNCTION OF ANNOTATIONS

03 Building an annotation system for the computer requires a clear outline on how annotations are used in the real world. Readers of any kind of paper or book annotate “as a routine part of their engagement with the material”^[1]. The reasons for annotating a document are numerous. Analyzing its form and function will expand the requirement list which is the basis for a digital annotation interaction scheme.

Catherine C. Marshall from the Xerox Corporation did a study in 1997 where she analysed the form and functions of students annotation in books. “I chose a university bookstore as the setting for this study, since it was a ready source of well-used books, uninhibitedly marked-up, in as many disciplines as we would find course offerings in a major university.”^[2] Looking at university books for annotations had several reasons. The most important out of the many is: “Examining books from the same edition ensures that the pages are the same in size, weight and texture of paper; that the margins are the same width; [...] that the books started out as materially the same”^[3]. That requirement made sure that the form and so its function was not disturbed by any material issues like different margin-space for example.

But what exactly is an annotation? The wikipedia defines: “Annotation is extra information associated with a particular point in a document. Most commonly this is used for example in draft documents, where another reader has written notes about the quality of a document at a certain point, ‘in the margin’.”^[4] Before the functions of annotations can be determined a form analysis is necessary. Using any kind of annotation tool (i.e. highlighter, pen, ...) demands a certain amount of attention. Warren Thorngate, a faculty member in the Psychology Department at Carleton University, Ottawa, Ontario, Canada summarized his book ‘On paying attention’ with the result that making choices expends attention. So highlighting text or even writing notes in the margin requires attention and drives the user away from the topic he/she is reading about. Catherine C. Marshall

observed in her study that “students who use highlighters write fewer marginal notes than students who underline passages with pens”^[5]. The reason behind that might be that it is far more uncomfortable writing notes with the thick top of a highlighter than writing these with a simple pen. Changing tools requires even more attention than simple highlighting and will take the user even more away from his/her reading activity. So a seamless interaction between reading and annotating is essential for the successful annotation of text. In one of the student books Catherine C. Marshall found a color coded highlighting scheme which definitely took a lot of attention to create and to consist and which is definitely not comprehensible for outstanding readers.

“The forms of annotation are also clearly shaped by disciplinary expectations and text book genre”^[6]. For example a pencil is the preferred marking tool in mathematics while complex philosophical narratives are subject to extensive mark-up in the form of underlines and highlightings. In english classes with difficult works in fiction read, students used the margin space quite intensively. Catherine summarized the annotations she found in a table which is presented visually interpreted in table 3.1.

Functions

“Annotations do not serve only a single function; they serve a multitude of functions”^[7] The Xerox Study defines six form&function categories. First, annotations are **procedural signals**, meaning that a student can easily overview where an assignments starts or which parts are important and which paragraph requires a second attention. Second annotations are **placemarks**; a summary of the annotations at the end of a semester for example will already create a rough sketch of somebodys work. Third annotations are acting as a **placeholder for problem-working** meaning that some people might annotate an area or write something in the margin where they solve a problem directly. They avoid an extra paper to not loose attention on the problem itself. Fourth annotations are **interpretations**, either from the reader or as a result of careful reading. Fifth markings can act as a **visible trace of a readers attention** and sixth any annotation can **reflect the circumstances** of reading which can be short comics or just random sketches which have obviously nothing to do with the text itself.

■ [1]

Catherine C. Marshall, Annotation: from paper books to the digital library, p. 131

■ [2], [3]

Catherine C. Marshall, Annotation: from paper books to the digital library, p. 132

■ [4]

Wikipedia, Annotation, 301812005

■ [5], [6], [7]

Catherine C. Marshall, Annotation: from paper books to the digital library, p. 135

	TYPE	CONTENT
TELEGRAPHIC	<ul style="list-style-type: none"> · Underlining · Highlighting · Circles and boxes around words and phrases 	<ul style="list-style-type: none"> · Brackets, angle brackets and braces · Asteriks and stars · Circle and boxes around whole pages · Arrows and other deictic devices to connect within-text markings to other marginal markings
EXPLICIT	<ul style="list-style-type: none"> · Brief notes written in between lines, especially translations of words in foreign language texts 	<ul style="list-style-type: none"> · Short phrases in margin · Extended notes in margin · Extended notes in blank pages in the front of the book · Problems worked in margins

► TABLE 3.1

Characteristics of annotations written in the student books

Annotations as procedural signals

Students annotate their material for further attention “to designate reading assignments, responsibility for ,knowing‘ and desire to re-read”^[1]. This function might result in a custom version of the text the student is reading right now. In a biology book for example Catherine C. Marshall found various crossed out reactions from a student. This does not mean that the student thinks that those reactions are not going to happen or are somehow wrong it just means that the student does not have to pay any attention to those ones while re-reading the book or summarizing the chapter.

Annotations as placemarkings and aid to memory

“Within-text markings that specify short extents (usually with highlighter, but sometimes with underlines or circles) seem to function as placemarkers, as a way of remembering or remembering to remember”^[2]. A closer look at the annotations in the biology book unveils that most of the time the highlighting took place at term definitions or other portions of text the student needed to memorize. The reaction itself was highlighted rather than the beginning or ending of the molecular configuration. In literature work the highlighting were connected to important passages and bits of dialogs which might be used later in a paper or essay.

Annotations as in situ locations for problem-working

“Students sometimes approach problems in context, at the time they are encountered, rather than deferring them and risking a break in attention or they work a problem near where the equation or theorem is presented in the text”^[3]. The biology book shows that one student used the figures there to work out molecular rotations or another student schematized atomic symbols in molecules to show evidence of counting marks. Before taking another sheet of paper or copying the existing one, students tend to draw directly in the margin or in the figures to solve problems they encountered in the actual paragraph.

Annotations as a record of interpretive activity

“Marginal notes, jottings and interpolations record interpretive activity”^[4]. The Xerox Study found three kind of interpretation in the margin

- Interpretation of unfamiliar language (old words, latin,...)
- Interpretations of structure (note names of frame scenes, ...)
- Interpretation of the work (i.e. a note in Richard III read: ‘Boy can't believe uncle is a killer’)

These annotations can be an insightful commentary and at the same time a documented misreading. Those annotation could affect the readers of the textbook since they

► [1], [2]

Catherine C. Marshall, Annotation: from paper books to the digital library, p. 136

► [3], [4]

Catherine C. Marshall, Annotation: from paper books to the digital library, p. 136

FORM	FUNCTION
<ul style="list-style-type: none"> · Underlining or highlighting higher level structure (like section headings) · Telegraphic marginal symbols like Asterisks · Crossouts 	Procedural signaling for future attention
<ul style="list-style-type: none"> · Short highlightings · Circled words or phrases · Other within-text markings · Marginal markings like asterisks 	Placemarking and aiding memory
<ul style="list-style-type: none"> · Appropriate notation in margins or near figures or equations 	Problem-working
<ul style="list-style-type: none"> · Short notes in the margins · Longer notes in other textual interstices · Words or phrases between lines of text 	Interpretation
<ul style="list-style-type: none"> · Extended highlighting or underlining 	Tracing progress through difficult narrative
<ul style="list-style-type: none"> · Notes, doodlings, drawings and other such markings unrelated to the materials themselves 	Incidental reflection of the material circumstances of reading

► TABLE 3.2

Mapping annotation forms into functions.

are quite personal and judgemental annotations and can be sublime helpful or silly misleading for the unbiased readers personal thoughts.

Annotations as a visible trace of the reader's attention

"Annotations become a visible trace of the readers attention when the material is difficult and in narrative form"^[1]. Philosophy texts for example where filled with highlighting or underlinings which documented in a very unique way the readers attention.

Annotations as incidental reflections of the material circumstances

"Markings are bound to reflect circumstances that are entirely outside of the realm of the text, fiben that there is a world of distraction outside of the book"^[2]. Catherine C. Marshall found on a page in a copy of the mathematics Calculus for example the sentence 'I love You'. It is quite obvious that the annotator was not referring to a function but rather to another person. Even drawings or patterns where found in the books.

So which requirements could be claimed from these findings?

Annotating, as every other action, requires attention. Since every computer interaction is by its nature serial it is really important to help the annotator to keep focus on his work. General computer ideas of overloaded toolbars or uncountable formatting choices will not help the user to focus on the reading process: Move eyes from the text to the toolbar - Decide which tool to use - Activate the tool - Set a color - Find the the text position for the marking again - Start marking. This would be a standard computer workflow ignoring the just observed findings. The focus lies on the reading and the user should spent the least amount of attention for annotating the desired document. Less choices can result in higher productivity:

REQUIREMENT 002

Simplicity. A computer annotation system needs to be fast, simple, direct and seamless.

REQUIREMENT 003

Changing a tool should be easy and fast.

Xerox Member Catherine C. Marshall finalizes her study with the words:

"The final implication – support for a smooth integration of annotating with reading – is the most difficult to interpret from a design point of view; yet it is potentially

► [1], [2]

Catherine C. Marshall, Annotation: from paper books to the digital library, p. 136

the most important. Until we, as System Designers, get this right, it is likely that people will continue to annotate paper materials, even as they read materials in a digital library"^[1]. This paragraph makes it quite clear where the focus on a computer based annotation system lies. Even today with a computer on nearly every desk people annotate or work with paper, which proves that Catherine C. Marshall's is right with her conclusion. Seamless integration is the key for a successful annotation system on the common computer systems.

Building a digital annotation system that serves all mentioned functions in this Xerox study can contradict the just postulated two requirements. Adding priorities to these functions to find a compromise between simplicity and function-serving could help to build a requirement list that allows an efficient interaction design. Each of the mentioned functions requires different interaction possibilities. Jotting comics is done differently than marking up paragraphs. These interaction differences will be responsible for the whole annotation experience and serving all of them will probably result in cluttered toolbars and nested menus. Something that would not fulfill the just observed requirement 002. 'Procedural signaling', 'placemaking and aiding memory' and 'tracing progress through difficult narrative' have mostly the same basic appearance and thus similar interaction to create. The common plain appearance could result in simple and attention focused interaction. Problem-working and interpretation could end up in quite complex and highly individualised notes, figures, equations,... which demands a more intense interaction with the computer. Thus problem-working and interpretation are lower prioritized in building the digital annotation system. Writing down notes in the 'Margin' or drawing basic symbols is part of the simple annotation process and should be possible with the digital annotation system. Giving the user tools for equation building or writing long texts will be hard to implement without losing simplicity. 'Incidental reflection of the material circumstances of reading' will have the lowest priority. These annotations are unrelated to the content and thus can be ignored.

REQUIREMENT 004

Simple Shapes, highlighting and short notes are part of basic annotation and serves most functions well.

■ [1]

Catherine C. Marshall, Annotation: from paper books to the digital library, p. 140

Catherine C. Marshall concludes her study with an own requirement list for an annotation system for a digital library: "*In situ annotation, distinguishable from the source. Readers like to write on the materials themselves – in the margins, in between the lines, over text, within figures.[...]*"^[2] This simple requirement is going to be a key difference in comparison to all digital annotation possibilities. In the next chapter we will see more evidence for this important feature.

REQUIREMENT 005

The annotation should be distinguishable from the source.

This basic idea was the starting point and main idea for this diploma thesis and will later be the solution for a lot of interaction problems other annotation systems have.

■ [2]

Catherine C. Marshall, Annotation: from paper books to the digital library, p. 139

04

Based on a study from at Xerox Research Cen-
Kenton O'Hara and tre (EuroPARC) in Cam-
Abigail Sellen conducted bridge.

READING MECHANISM ON PAPER AND ONLINE

04 The previous chapter showed that annotations are quite common in student books. People are used to do annotations on paper and as the study of Catherine C. Marshall found out these annotations serve quite different functions. Everybody uses different schemes on how to work with the current text. Since this thesis is about an annotation solution for current computer systems it is important to research on how differently people read their texts on paper or online. Xerox, popular for their paper serving products, did a study on this as well. Kenton O'Hara and Abigail Sellen asked ten volunteers from the research and administration staff of their Xerox PARC laboratory, to summarize a four page article from a general science magazine. Five Subjects were randomly assigned to the 'Paper' condition and the other five to the 'online' condition. The paper group needed to read and summarize the article on paper while the 'online' group needed to work fully digital.

The 'paper' group got the article, an extra paper for note taking and one paper for the final summary. In equivalent the 'online' group got the article in a Microsoft Word 6.0 document. Another empty sheet was opened with a headline called 'Notes' and another document was opened with the title 'Summary'.

After the task was completed the volunteers were heavily interviewed and the video taped sessions analyzed.

Findings from the 'paper' group:

The previous study about annotations in student books came to the same conclusion as the volunteers in this study: "[...] *Jannotating and note-taking while reading was important in deepening their comprehension of the text, and in helping them to form a plan for writing the summary*"^[1]. Markings were made quickly and interwoven with the ongoing reading. Two subjects were using the highlighter and pencil quite intensively (underlining, use of asterisks and notes in the margin). One subject went even further and devoted the thickness of a line as an indicator for the degree of importance to this annotation. Most of the subjects used the annotation as procedural signals. "*The first reading was quite slow and the second reading was skimming. The annotations helped me when I was skimming in the second reading... Whenever I finished writing about some point I skimmed forward - I looked for the next annotated bit*

■ [1]

Kenton O'Hara & Abigail Sellen, A Comparison of Reading Paper and On-Line Documents, p.337

and then i just read around it a little bit if I needed to remind myself of what they were trying to say"^[2]. One subject mentioned something interesting: "*as you underline something you re-read the words, and this enforces it more*"^[3]. This an important point as a result for the annotation function—procedural signaling. Through underlining or highlighting the reader is re-reading the current sentences/word/paragraph and this does not only serve the functions mentioned before, the item get even more emphasized in the readers mind since he/she reads it one time more.

Findings from the 'online' group:

Four of the five subjects expressed that if they would have done this on paper they would have used annotations. Only one subject attempted to do annotations online. He customized himself the word toolbar and tried to use the styles and shapes from the paper world in Microsoft Word. "*So the annotation was not as easy as all that [...] I think the whole process would have been a lot quicker on paper. Annotations are that much more flexible because you can write in the margins which you can't very easily do here. You have to establish a new text and then have to write*"^[4]. Another problem the online group mentioned: By annotating in Word, the original document gets modified. By italicising, emboldening or underlining specific words it is hard to distinguish between the original document formatting and the annotations addition. The subjects indicated that they would wish a direct separation of the annotation layer and the original content. Without this distinction annotating the document felt uncomfortable. As a result of not annotating online four subjects decided to use the note-taking document in Microsoft Word. Two of the subject used heavily the copy&paste command to fill the note-taking document. The other two subjects took most of the notes after the first reading of the paper, "*producing a plan almost entirely from memory with very little reference back to the source document*"^[5].

■ [2]

Kenton O'Hara & Abigail Sellen, A Comparison of Reading Paper and On-Line Documents, p.337

■ [3]

Kenton O'Hara & Abigail Sellen, A Comparison of Reading Paper and On-Line Documents, p.337

■ [4], [5]

Kenton O'Hara & Abigail Sellen, A Comparison of Reading Paper and On-Line Documents, p.337

Summary of the findings:

- Annotations on paper were effortless and smooth while online annotating distracted the reader and was somehow awkward.
- Flexibility of paper supported the annotations from the subjects while the online group felt uncomfortable annotating the Microsoft Word document and in addition they did not want to change the original document but rather have a separate layer for visual annotations.
- Both groups took heavy use of the note-taking sheet. While the paper group used this paper during the reading process, the online group took the notes after reading the text or was quite long distracted during the reading for note-taking.

To analyze the difference between paper and online documents it is interesting to see how the subjects reported the 'movements' between and within documents. Through the interviews Kenton O'Hara and Abigail Sellen figured out that there are three main reasons for spending a great deal of time moving through documents.

- a) Planning: To get an overview/plan of the text, it was important to move the pages around. Reading bits of every page is serving the reader with a quite rough summary and category for the kind of text.
- b) Reference: Subjects were scanning the text for specific quotes or spelling problems.
- c) Checking Understanding: Several times the people were re-reading some paragraphs to make clear they understood it right or to clarify doubts.

The paper group movements through paper was characterized by its speed and automaticity. Page turning is a good example for this, with one hand the subjects were lifting the page already, even before they finished reading, just to minimize the disruption between reading the text at the end of one page and the beginning of another. What makes this speed and flexibility possible is the use of two hands. In the real world people can easily use one hand in a pile of paper to search for a specific information. There are so many other paper interactions which involve the use of both hands but listing all of them would break the frame here.

Another important paper feature is *"the fixity of information with respect to the physical page"*^[1]. Meaning: *"I knew it was on the 3rd page, cause I could remember*

■ [1]

Kenton O'Hara & Abigail Sellen, A Comparison of Reading Paper and On-Line Documents, p.339

that it was in the middle [column], cause it was this botanix bit so I knew it was on this page"^[2]. Everybody is familiar with this feature and is quite conscious about it. Shuffling through a pile of papers and recognizing its content by the visual form of the text is no problem for the human mind.

Moving within or between online documents is quite different as the observation and interviews resulted. Scrolling or paging through an electronic document was found to be *"irritatingly slow and distracting"*^[3]. *"I was getting very annoyed and clicking on those things and shouting at it ... I just found that it took ages and ages. I was losing interest. It was distracting me from the point"*^[4].

One fact which limited quick movements in online documents was one-handed input. Actions in a computer are done serially. So in direct comparison to paper documents where two-handed interactions are natural the whole computer experience felt awkward and it was nearly impossible to perform anything concurrently.

Spatial properties in computer interfaces are making flexible movements through documents not easier. Grabbing a piece of paper to move it around or using a thumb for shuffling through a stack of paper is impossible on current computer systems. The user uses window title bars to grab an electronic document and move it around. The space of a title bar is in comparison to a whole sheet of paper unequally smaller. Today's interaction designers know that switching between documents inside one application or even between different applications is a common computer interaction. Tabs for example are a popular design concept to organize windows of one application on the computer screen. *"A tab view is a convenient way to provide information in multiple pages. A tab view usually contains a row of tabs that give the visual appearance of folder tabs. When the user clicks on a tab, the tab view displays a view page provided by your application"*^[5]. Another popular solution for making moving between documents easier is a function called Exposé in Mac OS X. Evoking Exposé gives the user a thumbnail view of his windows and clicking on the appropriate thumbnail makes the window active. From an

■ [2], [3], [4]

Kenton O'Hara & Abigail Sellen, A Comparison of Reading Paper and On-Line Documents, p.338

■ [5]

Apple Computers Inc., Apple Developer Connection: User Experience/ Controls & Cells, Chapter: TabView

interaction point of view handling of documents in the computer and in the real world is very different.

Some paragraphs earlier it is stated out: "*The fixity of information with respect to the physical page*"^[1]. The position of information in respect to its visual appearance can be locked as well (zoom out, layout view,...). But does this already provide the fixity of information? The subjects of the online group denied that question. First, to look at the complete page, Microsoft Word 6.0 offers a function called layout view. It is impossible to read any of the information on it since the typography is very small and the screen resolution too low to recognize characters. By zooming into the document, the user will reach a point where he/she can read everything but sooner or later scrollbars will appear and at that point the visual fixity is lost. Today's computer displays are too small or have a too low resolution to display two documents next to each other which could solve the fixity issue. So the mechanism of remembering information by its position on the document is not valid for the computer experience right now.

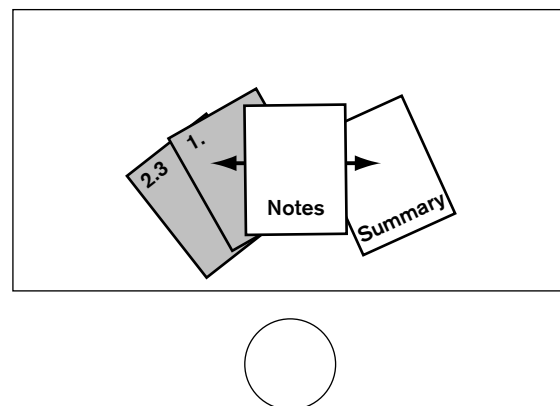
In direct comparison to the paper group it is really obvious how serialized the workflow in the computer is. The subjects were reading their documents, then grabbing the note document and writing down some ideas, then re-reading specific passages and copy&pasting items to the note-document and then building the summary. The paper group did such things parallelly. While reading they annotated some text or wrote notes on the note sheet. This complete different interaction is hard to conclude into one specific requirement but it is an obvious difference and needs to influence the whole interaction architecture of an electronic annotation system. Copy&paste became the digital version of note taking in the real world. These two commands demand switching between two windows on a computer and as mentioned earlier, moving between windows can be attention consuming. A seamless and fluid annotation system could suffer from the interaction behaviour of the copy&paste command. It is a common action performed by all computer users. Any adjustment to those two basic commands must be done in the context of overall consistency. Improving this interaction is not a requirement but something which could result in a more efficient workflow. Since copy&paste is the digital version of note

taking in the real world, a digital note taking possibility could avoid the time and attention consuming interaction of moving between documents.

Switching documents is closely connected to the topic 'spatial layout of computer interface elements'. The subjects were laying out their documents on the table quite differently. The reason for positioning the given documents in a specific order could be summarized as follows:

- To gain a sense of overall structure
- To cross-reference
- To interleave reading and writing

Especially the subjects of the paper group where using this intensively. While writing the summary they moved the pages around. The whole spatial appearance was very dynamic. Here is one example of such a layout



■ FIGURE 4.1

An example of document placement layout during the planning phase of writing

The Subjects of the online group where quite limited concerning moving pages around:

"It was annoying that it's got to be the current window in order to be able to move, and the current window is always the one in front. So I can't hide the one I was typing into behind there and simply move the pointer into it and start typing or pasting or whatever"^[2]. A good way to experience this limitation is to summarize a given text while writing the summary on the backside of the given paper. This flipping of the page shows how important the attention focus of a digital annotation system is.

■ [1]

Kenton O'Hara&Abigail Sellen, A Comparison of Reading Paper and On-Line Documents, p.339

■ [2]

Kenton O'Hara&Abigail Sellen, A Comparison of Reading Paper and On-Line Documents, p.340

05

The following chapter is based on a study conducted at the IMF. It researches the way employees are working with paper and electronic documents.

RELATIONSHIP BETWEEN PAPER AND ELECTRONIC DOCUMENTS

05 The previous chapters showed that annotations are a common tool to support reading and working with texts. But how much do people annotate in their daily life. How often is the paper used in comparison to the electronic document and for what purpose? What role does an annotation play here? A study from EuroPARC, a Xerox Research Center, directed a study that answers some of these questions. Abigail Sellen and Richard Harper “report on an examination of work practice in a knowledge-based, document-intensive organisation and describe the role of paper in that world”^[1]. The study took place at the IMF (International Monetary Fund) based in Washington DC, USA which the wikipedia defines as: “The International Monetary Fund (IMF) is the international organization entrusted with overseeing the global financial system by monitoring foreign exchange rates and balance of payments, as well as offering technical and financial assistance when asked”^[2]. The IMF has 3.000 employees with 900 professional economists which all originate from different countries. Some reasons for choosing the IMF:

- The work at the IMF is document intensive
 - As in many other offices the IMF uses electronic and paper documents in a mixed environment.
 - The IMF analysis its own work practices quite intensively to increase efficiency and invests quite an amount of money in future information technology.
- Eight economist (desk officers and chiefs) provided detailed information on how they use paper on a daily basis. At the end of each day they were interviewed to get exact durations and actions which were connected to the submitted log.

As stated out before, the IMF work is quite document intensive and so it is not surprising that “97% of their time was spent on activities which involved documents of some sort”^[3]. From those activities “87% of the time was spent on activities involving papers: 51% of the time they involved only paper documents, 35% of the time they involved a combination of paper and elec-

tronic documents, and 14% of the time they involved electronic documents only”^[4]. The diagram on the next page is a visual interpretation of a figure presented in the study documentation. It shows a more detailed view of the daily activities from the IMF economist which were collected throughout the study.

The daily work at the IMF relies on a great amount of paper or a combination of paper and electronic tools. Further this profile shows that a large amount of their time was spent on authoring activities. An interesting fact is that in collaborative authoring, either in co-authoring a document or in reviewing the documents of others, the authoring was heavily paper-based. Drafting and editing of own texts or data was mostly a mixture between paper and/or electronic documents or even electronic processing only.

The figure also shows that over half of the conversations or meetings were supported by paper and it also seems that paper is the preferred medium for reading documents and/or delivering as for thinking and planning activities and for document organisations.

This daily activity study shows that 71% of the authoring time was spent for editing and reviewing, while 29% of the time was spent drafting documents. The drafting was mostly done alone while editing and reviewing was a mixture of individual or collaborative work independently if it was their own or the documents of somebody else. Certainly the IMF is somehow special in the way the institution deals with documents since nearly all documents are co-authored and there is a complex structure how to negotiate all facts in a document with.

“The activity analysis also showed that much of the editing and reviewing process involved the review of other people’s documents. This was always done by marking up the document on paper”^[5]. One property of paper is the distinction between the reviewer’s comments and the original text. This is true in two senses: First there is a true separation between suggested modification and actual modification. The distinction between the annotation and the original document leaves the actual implementation to the original author and owner of the document. He/she can easily review suggested

■ [1]

Abigail Sellen & Richard Harper, Paper as an Analytic Resource for the Design of New Technologies, p.319

■ [2]

Wikipedia, International Monetary Fund, 30|10|05

■ [3]

Abigail Sellen & Richard Harper, Paper as an Analytic Resource for the Design of New Technologies, p.321

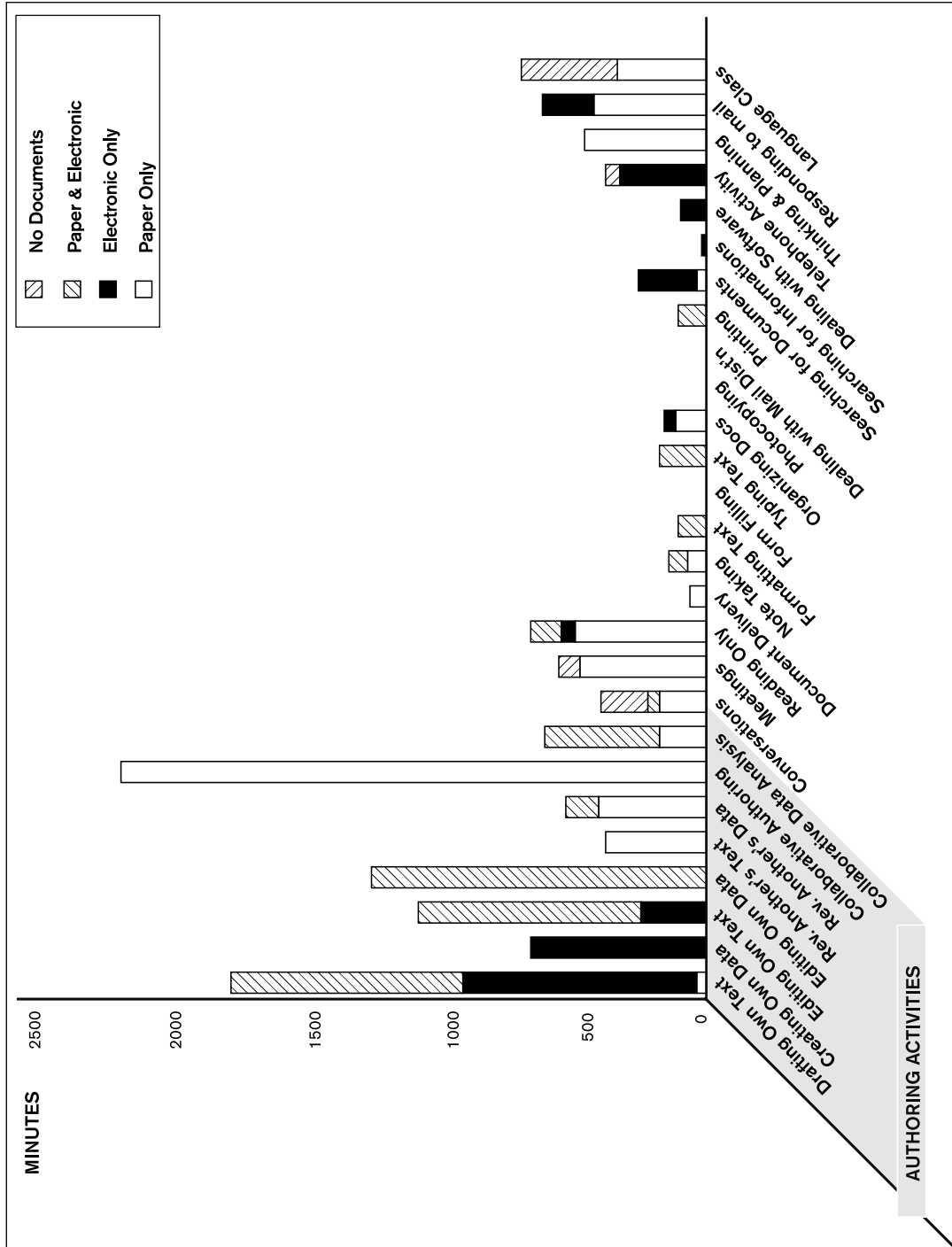
■ [4]

Abigail Sellen & Richard Harper, Paper as an Analytic Resource for the Design of New Technologies, p.321

■ [5]

Abigail Sellen & Richard Harper, Paper as an Analytic Resource for the Design of New Technologies, p.322

FIGURE 5.1
The activity profile for
eight desk officers and
chiefs at the IMF.



and actual modifications. Secondly, this distinction is a perceptual one. It is easy to distinct between the original text and the comments. This provides the final author or owner of the document with an overview of the situation at a glance.

Another feature of paper compared to actual electronic annotation systems is the “*richness of representation*”^[1]. In environments where more than one person provides comments to one document it is important to see which annotations belongs to whom. On paper this is easily achieved by the nature of markings on it. It is done in many ways: colors, handwriting, different pens and also by adding text, graphics and sketches of tables for example. A kind of markup language evolved on how to indicate that a paragraph needs to be moved, deleted or otherwise modified. Most of the current electronic annotation systems do not provide the user with such a flexibility and rich set of simple and natural tools.

What are the results from this study concerning the digital annotation system?

- Paper still plays a major role in today's office environments. Even a greatly structured institution like the IMF which is open to new information technologies uses paper quite intensively.
- Drafting documents is well balanced between paper and electronic documents. The current computer technology obviously provides a good set of tools for creating documents.
- Editing and Reviewing: The study subjects spent 71% of their daily authoring time for these activities and they were highly depending on paper especially while working with documents from colleagues. So the computer plays a secondary role here.

The participants from the previous Xerox Lab study called for a distinction between the annotations and the original content. But that was just a claim for a visual separation since the participants were italicizing, bolding, underlining annotated words or paragraphs. A visual distinction of the original material and annotated content could not be made anymore. This study demands even more—a distinction of annotations by authors. This introduces another layer of abstraction to the interaction design of the annotation system. The distinction by authors also implies the issue of versioning (i.e. who annotated

first) and structural issues (i.e. visibility of some authors). Paper fulfills these requirements easily through its basic properties (individual handwriting and drawing skill and different pen colors and types). On a digital annotation system these simple requirements could threaten the requirement of a seamless and fluid integration.

REQUIREMENT 006

Annotations need to be distinguishable by its content and author.

Another topic this study has brought up is the aspect of collaborative working. The social aspect of paper has not been mentioned yet but is particularly in the computer world a very important chapter. Most of the editing and reviewing at the IMF was done in collaborative situations and especially in those situation the electronic solution was inferior to the old paper. Collaborative authoring was done only on paper. Data exchange is a basic requirement in a working environment and the social aspect of exchanging data is normally lost by sending documents via email or putting files on a server. Today's computers are connected to millions of other computers nearly out of the box. The internet supports the possibility of sharing annotations or annotate documents together in an electronic environment. The next chapter is going to deal with those ideas.

■ [1]

Abigail Sellen & Richard Harper, Paper as an Analytic Resource for the Design of New Technologies, p.322

06

This chapter is based on a study initiated from the Microsoft Corporation. It analyzes if annotations undergo changes when they are made public.

FROM PERSONAL TO PUBLIC ANNOTATIONS

06 The previous chapter was about the usage of paper and electronic documents in daily activities at the IMF (International Monetary Fund). As a result it was quite a surprise that especially in the area of reviewing and editing documents, collaborative authoring was the main issue.

This chapter explores the relationship between personal and public annotations. Catherine C. Marshall who is currently working for the Microsoft Cooperation and who did the already mentioned study on 'Annotation: from paperbooks to the digital library' for the XEROX Parc Institute, is doing another study with the help of A.J. Bernheim Brush (Department of Computer Science and Engineering/University of Washington) about the topic of public and private annotation. She and her colleague analyzed over 1.700 annotations and stated out that only a small portion of these annotations were made public and while those personal annotations were made public they went through dramatic changes in its appearance and even in its content. The study uses three sources of data—interviews with the participants, a collection of their annotations and finally their contribution to online discussions and summaries. The eleven students who took part in this study were participating on a Human-Interaction seminar. Their task was to read three selected papers each week which were provided on a class website. The students were asked to write a brief summary of the texts and to participate at online discussions for each paper. They were using a tool called WebAnn (Figure 6.1) to write the shared summaries. All in all this study took

four weeks and twelve texts were used for the final findings. Catherine C. Marshall collected 49 readings from the students. Two of the eleven did not hand out their readings and another two students did not annotate their paper while they were reading.

Creating an annotation in WebAnn is done by highlighting a specific text and associating a text to the selection in the left pane. Others can reply to this text which is visualized in a hierarchical form. The students could also add summaries of the whole text at the end of each page.

In the study analyzed earlier, Catherine C. Marshall came up with a quite complex system of form and functions of annotations. 5 years after the Xerox study she presents in this Microsoft study a much more compact form scheme of annotations which is visualized in table 6.1 on the following page.

This system allows better defined categories and allows a faster analysis of the 1.700 annotations. She divided the annotations into the categories: Anchor-only, Content-only and Compound while a compound consists of more than one type. For example an underline plus a note:



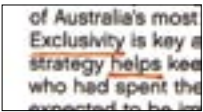
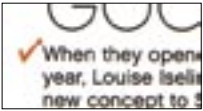


FIGURE 6.1 WebAnn embedded in Microsoft Internet Explorer

ANNOTATION TYPES	
ANCHOR ONLY	<ul style="list-style-type: none"> · Underline · Highlight · Circle · Margin bar
CONTENT ONLY	<ul style="list-style-type: none"> · Note · Mark (e.g. *) · Other (e.g. doodles)
COMPOUND	<ul style="list-style-type: none"> · Anchor + content · Complex anchor · Complex content

TABLE 6.2 The annotation type categories

Online annotations in WebAnn are always a compound of the subtype anchor+content since an annotation consist of a highlight and an associated note.

The submitted papers (online or paper) were researched by 'extent' (selection length) and 'structural relationship' with the categories: word, term, phrase, multiphrase, sentence, paragraph, list, heading, document and unanchored.

ANNOTATION EXAMPLE	TYPE	ANCHOR TYPE	CONTENT
	Underline (Anchor Only)	word	none
	Mark (Content Only)	sentence	
	Anchor+ Content (Compound)	phrase	"buy it"

► TABLE 6.1

Examples of coding for personal annotations

The comparison of personal (paper) and public (online) annotation was easy for content related notes. For example one person wrote

on paper (personal): "similar to cue cat; failed"^[1]

online (public): "This sounds similar to the cuecat device distributed by Radio Shack free of charge. More information can be found here [URL]. As I recall it hasn't been widely successful but some users really like it"^[2].

In the anchor only comparison the correspondence was much harder to determine. If the WebAnn annotation was close to the paper annotation Catherine C. Marshall declared a correspondence.

How did those 1.700 paper annotations make their way into the public discussion? The interviews with the participants about their reading and annotation practices made clear that nobody read the given text online. Even though they were available on the course website every subject printed out a copy to follow the assignment on paper, independently if he/she used annotations or not. This fact has been discussed earlier in this thesis while referring about the study from Kenton O'Hara and Abigail Sellen. But the students in this study bring up a fact that has not been mentioned yet:

By making the annotations on paper they were automatically saved and associated with the right document. One student went even further and expressed that annotations "saved her readings because she's 'taken the time to write stuff on them'"^[3]. This is an important fact

which needs to be considered while designing the behavior of a digital annotation system.

As mentioned in chapter four the students were explaining their decision to read on paper with the aspect of fluidity. The subjects who annotated the texts did that during the reading process even though they had an online annotation system available. WebAnn did obviously not translate a multitask action (annotating while reading) into a serial action (reading->annotating->reading->annotating...).

Since this study explores the difference between personal and public annotations, it is interesting to see if the assignment of publishing personal annotations in WebAnn changed the way the participants annotate. One student who characterized himself as a 'highlighter' took some simple notes in the margin to remind himself what to say in the online discussion. Other students were annotating their texts with an eye towards their colleagues. Another student distinguished between notes for herself and notes for the public and others would actually just annotate what they would like to contribute later. So the assignment or just the simple awareness that some of their annotations are going to be public, changed the way of their annotating practice.

But this did not change the way personal annotations were created—an unconscious act done while reading. Some students were quite surprised while the interview analysis of how much they actually marked up the text. One student went even further and said: "I don't know why I wrote some of the things I did"^[4]. This again

► [1], [2], [3]

Catherine C. Marshall&A.J. Bernheim Brush, Exploring the Relationship between Personal and Public Annotations, p.352

► [4]

Catherine C. Marshall&A.J. Bernheim Brush, Exploring the Relationship between Personal and Public Annotations, p.352

proofs how important the seamless integration has to be.

WEEK	PERSONAL ANNOTATION (on paper)	PUBLIC ANNOTATION (online)	TABULATED PAPERS
4	613 (86%)	97 (14%)	17
5	236 (79%)	64 (21%)	12
8	521 (89%)	64 (11%)	13
9	165 (86%)	27 (14%)	7
TOTALS	1535	252	49

■ TABLE 6.3

Overview of materials collected.

Each week, three readings were assigned.

This table summarizes the ratio of public annotations in relation to the personal ones on paper. Students created a lot more annotations while reading than they finally shared. This determines the results from the interviews with the subjects.

Table 6.4 shows an overview of the personal annotation types. Most of the personal annotations are from the category 'anchor-only'. The previous study from Catherine C. Marshall showed that people who are underlining are also pretty active in note-taking. Looking at papers with a lot of text highlighting shows less notes in the margin. So the claim that changing tools requires attention and that this might be the reason why mostly one-tool annotations are present, is re-proofed here.

Less than 20% of the overall notes were content-only or even compounds which are called traditionally marginalia. This observation is interestingly similar to a study from Barger and Moscovich called 'Reflowing Ink-Annotations'. This study will not be further examined since their task/approach and findings were rather in a technical context. They studied the redrawing and linking of annotations in relation to content from a mathematical point of view and build a framework to control the reflowing of free form ink annotations. It is interesting to see the same results from two independent studies.

Catherine C. Marshall examined that nearly half of the personal annotations (641 or 45%) were made at a „sub-sentence“ level. Only 402 or 26,2% were at a sentence level and 263 or 17,1% at longer segments of text. The

meaning of this for a digital annotation system will be discussed in the summary to this text.

As mentioned earlier in this text, public annotations in WebAnn are always a compound of anchor+content. Thus is an analysis realized with the personal annotation not possible. Catherine C. Marshall observed that from the 252 public annotations 48 or 19% were part of the shared summary and 204 or 81% were part of the online discussion—a significant difference. From those 204 online discussion annotations 121 were actual comments and 83 were replies. While personal annotations were mainly at sub-sentence level, 42% of the shared annotations were anchored to sentences. 19% were anchored on words or phrases. *“A quality examination of these anchors shows them to be on more carefully selected portions of text”*^[1].

ANNOTATION TYPE	NUMBER	FREQUENCY
ANCHOR ONLY	1262	82,2%
Underline	842	54,9%
Highlight	250	16,3%
Circle	140	9,1%
Margin Bar	30	2,0%
CONTENT ONLY	120	7,8%
Note	83	5,4%
Mark (eg. *)	21	1,4%
Other (eg. doodles)	13	1,1%
COMPOUND	153	10,0%
Anchor + content	138	9,0%
Complex anchor	14	0,9%
Complex content	1	0,1%
TOTAL	1535	100%

■ TABLE 6.4 Overview of personal annotation types

The previous findings lead Catherine C. Marshall to three hypotheses about the relationship of personal and public annotations:

- a) *“only a small fraction of the personal [...] annotations are shared with others”*^[2]

■ [1], [2]

Catherine C. Marshall&A.J. Bernheim Brush, Exploring the Relationship between Personal and Public Annotations, p.353

ANNOTATION TYPE (on paper)	ANNOTATIONS ON PAPER (total)	CORRESPONDING PUBLIC ANNOTATIONS ONLINE		
		Total shared online	Number used in online summaries	Number used in online discussion
ANCHOR ONLY	1262	247 (19,6%)	205 (16,2%)	42 (3,3%)
Underline	842	167 (19,8%)	135 (16,0%)	32 (3,8%)
Highlight	250	46 (18,4%)	38 (15,2%)	8 (3,2%)
Circle	140	31 (22,1%)	29 (20,7%)	2 (1,4%)
Margin Bar	30	3 (10,0%)	3 (10,0%)	0 (0,0%)
CONTENT ONLY	120	44 (36,7%)	21 (17,5%)	23 (19,2%)
Note	83	42 (50,6%)	20 (24,0%)	22 (26,5%)
Mark (eg. *)	21	2 (9,5%)	1 (4,8%)	1 (4,8%)
Other (eg. doodles)	13	0 (0,0%)	0 (0,0%)	0 (0,0%)
COMPOUND	153	88 (57,5%)	33 (21,6%)	55 (35,9%)
Anchor + content	138	84 (60,9%)	29 (21,0%)	55 (39,9%)
Complex anchor	14	4 (28,6%)	4 (28,6%)	0 (0,0%)
Complex content	1	0 (0,0%)	0 (0,0%)	0 (0,0%)
TOTAL	1535	379 (24,7%)	259 (16,9%)	120 (7,8%)

■ TABLE 6.5

Correspondences between personal annotations (on paper) and public annotations (online).

- b) *“some types of annotations are more likely to be shared than others and some annotations styles are more readily shareable”*^[1]
- c) *“annotations made while reading undergo profound transitions when they are made public; both their content and the extent of their anchors are changed to make the annotations intelligible to others”*^[2]

Table 6.5 displays in detail the correspondence between personal and public annotation and the corresponding public annotations are further broken down by online use in summaries and as commentary. This table will prove the correctness of Catherine C. Marshall's hypotheses.

Looking at the bottom row shows that from the 1535 personal annotations the students only made 379 public. This examination supports the first hypothesis: only a small amount of annotations are shared. From those 379 public annotations were 259 related to collaborative

authoring (shared summaries) and just 120 annotations were comments which students used in the online discussions.

The table also gives a good overview of which annotation types are shared. The most common shared annotations are notes and anchor+content. If an annotation consist out of content it is most likely that this annotation will be shared. This is also examined throughout the interviews where the subjects reported that they were writing down notes which should remind them of what they wanted to contribute online. So the compound subtype anchor+content is the mostly shared annotation type from this study. Anchor-only annotations were the least likely to be directly transferred from personal to public annotations. This category demands more attention since the effort of interpreting the markings in a collaborative situation is much higher. Only 3% of the anchor-only comments were used on online discussions but anchor-only annotations are quite helpful for summaries (16% were used in shared summaries). That leads to the conclusion that in this study anchor-only annotations are much more personal than content+anchor annotations.

■ [1], [2]

To see how the students changed their annotations because they put them online, Catherine C. Marshall examined content changes that occur when annotations are shared. This was done by examining the 120 shared annotations from the online discussion (Table 6.5 last column). She made 5 categories to document the changes:

- cleaned up
- original and more
- cryptic to understandable
- nothing to something
- unrelated.

Only 8.3% were cleaned up while most of the personal annotation needed to be extended before they make their way to the public. Nearly half of the personal annotations (43,3% needed to be added with content because no notes were made while reading).

This detailed study contains a lot of new information which will help in building a digital annotation system. But it also confirms some of the findings from the previous studies. The first remarkable fact is that the students choose to read on paper again. A digital annotation system will not change this fact since the reasons for reading text on paper are numerous. Spatial advantages, simplicity and papers support for social mechanisms are superior in comparison to electronic documents. The intention of a digital annotation system should not be to vanish paper or to bring the paperless office closer. The system could/should increase efficiency on how people work with digital documents and would/should help documents handling in their daily computer and office life.

One student in this study mentioned an important fact which is always taken for granted but is a requirement which definitely needs to be taken into consideration. It is the fact that annotations on paper are saved directly at the time while the user writes them down. He/she should not need to remember to save the annotations to avoid instant data loss. That would be a real drawback to a digital annotation system.

REQUIREMENT 007

The annotation system needs to auto-save markings and notes.

The majority of annotations in this study are, anchor only annotations like underlines or highlights. 82,2% or 1262 annotation are anchor-only. Isolating this fact, it is logical that a digital annotation system should have its first hand focus in providing the possibility of anchor-only annota-

tion. Simple highlighting or underlining should be a key feature. This statement gets even more important by taking a look at the structure of annotations. Personal annotation were mostly annotated at sub-sentence level(45%) which is a good indicator for fast unconscious jotting down while reading. When the students revisited their texts and prepared it for publicity most of the annotation where anchored with complete sentences which shows a thoughtful and conscious work through the text.

REQUIREMENT 009

A key feature is the support of fast and unconscious anchor-only annotation while reading.

The study about the IMF was oriented on collaborative authoring. This study concluded that only a small fraction of annotations are shared. In detail just 7,8% from the personal annotations on collaborative authoring. On the first hand this looks like a big contradiction to the findings from the IMF study. But, first the previous chapter focused on a comparison of the amount of paper versus digital documents used in their daily life and not about the amount of annotations on it. Secondly, the students had to use WebAnn as an annotation system which did not translate the multitask workflow into a serial workflow very well. And third these two studies took place in different environments. The IMF is a highly organised institution while this study observed a loose student class.

The fact that only a small fraction of annotations were shared is very interesting. Personal annotations were dominated by anchor-only annotations while WebAnn only allows Anchor+Content annotations. This fact could influence the way the students annotated. Personal annotations had a high level of rewriting and rethinking and underwent profound changes on their way to the public. The annotation structure in WebAnn is different to the majority of the source annotations. Publishing anchor-only annotations in WebAnn requires additional notes/text.

REQUIREMENT 010

Personal (local) and Public (non-local) annotations needs to behave and work exactly the same way. WYSIWYG. (acronym for 'What You See Is What You Get')

07

Can annotations effect an unbiased reader of a given text?

This text is based on a study from the University of Texas at Austin.

EFFECT OF ANNOTATIONS ON STUDENTS

07 In the previous chapter Catherine C. Marshall studied the relationship of personal and public annotations. So far by looking at the functions annotations serves the research always assumed that the reader of the annotations has also been the author of them. This study is going to take a closer look on the effect foreign annotations could have on an unbiased reader. Could those annotations change a readers opinion or attitude towards the text? Digital data is easy to copy and transfer. A digital annotation system will be no different. But will this obvious feature have a positive impact on an unaffected reader? Annotations are highly subjective. The form is not closely connected to its function which makes it impossible to interpret annotation by their appearance. The unbiased reader reads the content always together with the annotation. Does it make a difference if the reader knows the annotation author?

Annotations are due to its unconscious creation highly individual in its form and expression. How to measure the influence of an annotation? Joanna L. Wolf from the University of Texas at Austin asked for her study 'Effects of annotations on Student Readers and Writers' 123 un-

dergraduated students to write a persuasive essay based on a given text. After writing these essays the students had to work through a questionnaire which asked to recall information that was annotated, their overall attitude towards the text and evaluations of individual claims from the given texts. The written texts were analyzed in form of quality and overall position on the controversy.

Nearly half of the 123 students were first year students enrolled in an introductory composition course and the other half were enrolled in an advanced composition courses. This study took place as a reading and writing exercise during those regular composition courses.

The texts were about "a series of letters to the editor of the New York Times responding to a 1997 controversy by the protest of five Orthodox Jewish students against Yale University's policy requiring all students to live on-campus for their first two years of attendance at Yale."^[1] This issue was chosen because it is a text where students are potentially interested in and where a concrete attitude could be expressed. Before handing out the texts the instructors had to annotate the texts to define six 'hot-spots'(Table 7.1):

DESCRIPTION OF PASSAGES IN PRIMARY TEXT THAT RECEIVED ANNOTATION.	ANNOTATION FOR EVALUATIVE 1(+/-)	ANNOTATION FOR EVALUATIVE 2(-/+)
1. By Elisha dov Hack opposing Yale's dorm policy and criticizing Yale's "anything goes" residential religion.	"great metaphor", "clear presentation of the claim" (+)	"unfair", "distortion of Yale's policy" (-)
2. By Jewish student describing his positive experience at Yale and claiming that unlike Hack, he is glad to be here.	"is this true? ouch!" (-)	"great ethos! yes!" (+)
3. By Dean defending Yale's policies and expressing willingness to accommodate students.	"sounds like Yale is doing all it can", "works for me." (+)	"specifics? sounds like typical administrator ,b.s.'" (-)
4. By a law professor critiquing the irony of Hack's desire to return to the residential policies of the 1950's when Yale had a quota limiting the number of Jews admitted.	"wow. great point.", "enforced segregation in the 50's.", "self-segregation now." (+)	"but Hack doesn't argue that we should return to this policy." (-)
5. By a Texan opposing Hack and comparing his lawsuit to suing a restaurant for serving non-Kosher foods.	"logical flaw" „bad analogy and possibly racist" (-)	"good analogy" (+)
6. By a New Haven resident suggesting that Hack circumvent Yale's policy by maintaining two residences.	"is this for real? harsh!" (-)	"I agree", "he has options" (+)

■ TABLE 7.1

Description of the Passages and the Annotations they received in the two Evaluative Annotation Conditions.

■ [1]

Joanna L. Wolf, Effects of Annotations on Student Readers and Writers, p.20

These six hot-spots lead to four different letter versions:

Version A

No Annotations—'Clean' texts without any annotations were handed out.

Version B

Underlinings only—The six 'hot-spots' were underlined only. No comments in the margin.

Version C

Evaluative Annotations 1 (+,-,+,-,+,-)—These letters had evaluative commentary in the margin at the six 'hot-spots'. Three of those evaluative were positive the other three were negativ. The first comment was positiv, the next negative, the following two were positive again and the last two negative.

Version D

Evaluative Annotations 2 (-,+,-,-,+)—These letters are almost the same like 'Evaluative Annotations 1' just that the pattern of commentary changed to shift the overall attitude of the annotations. First was negative followed by a positive commentary while the next two were negative again and the last two positive.

These four different versions were randomly distributed to the 123 students together with a blank sheet of paper, an instruction page and a demographic questionnaire. Students with annotated letters had a note on the instruction page saying that these annotations were from a composition instructor who used those letters in a different class. The students had 35 minutes time to read the letters and write their essays. When the students handed out their work, they received a second questionnaire asking the students to recall some annotated and some 'clean' information from the text and general orientation to the controversy.

The results from this study are presented in a highly statistical manner which means that quoting them in detail would exceed the limit here. The results can be categorized into four general questions:

- Did the attitude of the students got immediately changed by reading a specific annotation to a specific passage?
- Did the overall attitude got influenced through the markings?
- Did the annotation change the style of writing?
- Are there any significant differences between the text of group 'Evaluative 1' and group 'Evaluative 2'?

	ANNOTATED ITEMS	EVALUATIVE 1 GROUP	EVALUATIVE 2 GROUP
PASSAGE SET 1	1. By Elisha dov Hack opposing Yale's dorm policy and criticizing Yale's „anything goes” residential religion. 3. By Dean defending Yale's policies and expressing willingness to accommodate students. 4. By a law professor critiquing the irony of Hack's desire to return to the residential policies of the 1950's when Yale had a quota limiting the number of Jews admitted.	Positive annotations	Negative annotations
PASSAGE SET 2	2. By Jewish student describing his positive experience at Yale and claiming that unlike Hack, he is glad to be here. 5. By a Texan opposing Hack and comparing his lawsuit to suing a restaurant for serving non-Kosher foods. 6. By a New Haven resident suggesting that Hack circumvent Yale's policy by maintaining two residences.	Negative annotations	Positive annotations
PASSAGE SET 3	No annotations in any of the groups	—	—

■ TABLE 7.2

Overview of the passage sets between the two evaluative groups

To analyze the effect of specific claims throughout the texts the post-writing questionnaire asked students to rate the persuasiveness of three passage sets from the source material. Table 7.2 shows the structure of these passage sets.

Taking a look at the statistical analysis Joanna A. Wolf did of passage set 1, group evaluative 1, who received positive annotations in this set, rated the claims significantly persuasive and favorably than the other two groups. The evaluative 2 group rated the claims in this set persuasive as well but less favorably than the other groups. By looking at passage set 1 only, it seems that annotations effect the readers persuasiveness.

While looking at the statistical data of passage set 2 it shows that only students receiving negative annotations for this passage rated it more favorably than the evaluative 1 group which received positive annotations there or the group with no annotations or underlining. This might have its reason in the already high rated base persuasiveness of this passage set. Joanna L. Wolfe concludes these findings with the possibility *“that negative annotations may be generally more influential upon attitudes than positive annotations”*^[1]. The analysis of passage set 3 resulted in nothing significantly to mention.

But did the influence of the single claims change the overall attitude towards the source texts? The Evaluative 1 group got positive annotations on passage set 1. That allows the conclusion that the readers attitude will follow the annotators favoring position in this letter and therefore supports Elisha Dov Hack. The opposite is true. Over 90% of the students adopted the position of being opposed to Hacks lawsuit in their essays. Thus annotations did not seem to influence students overall attitude at all. Actually this does not seem surprising since both evaluative groups received equally the same biased annotations (three positive and three negative). It would be interesting to know what would happen in an unequally relationship of annotations.

What happened to the students essays? Is there an influence from the annotations which changes the way the text got summarized? The form and functions chapter in this thesis pointed out that annotations are *“tangible reflections of a reader’s engagement with the text”*^[2]. This statement allows the conclusion that reading annotated texts will help them understanding how somebody else approached this document. That conclusion might allow

■ [1], [2]

Joanna L. Wolf, Effects of Annotations on Student Readers and Writers, p.23

the thesis that the students add more of their *“knowledge, assumptions and values to a reading”*^[3]. Simple summaries are quite common in student essays and Joanna L. Wolfe was curious to see if the simple fact of annotated text could change this and support more personal texts. To examine this she divided the text in so called ‘T-Units’ and assigned to each unit a value like: *summary, exposition, argument, ethos, concession or irrelevant*.

In this thesis ‘summary’, ‘argument’ and ‘irrelevant’ T-Units will only be addressed :

“Summary: *Factual summary of material found in the source materials (for example: 5 students are suing Yale university)*

Argument: *direct or indirect statement or support of a position on the issue (for example: I agreee with Dean Brodhead that Dorm life is crucial to a Yale education)*

Irrelevant: *information unrelated to the argument or the writing task (for example: I really need a tissue right now)”*^[4]

Analysing Joanna L. Wolfe statistical data shows that people from the ‘no annotation’ group wrote essays with the greatest amount of summary (11%) while subjects from evaluative 1 group had the lowest amount of summary in their texts (4,4%). At the same time this group had the highest amount of arguments (81,5%). These results allow the conclusion that the audience, a reader is virtually writing for, is going to change through the influence of annotations where the author is unknown. All groups who had annotations in their texts wrote essays with less summaries than the ‘no-annotation’ group.

The texts from the two evaluative groups differed quite significantly. The evaluative 1 group had the highest amount of arguments in their texts. In comparison, the evaluative 2 group used the least amount of arguments in their essays but the highest amount of ‘T-Units’ from the category irrelevant. Joanna L. Wolfe concludes: *“Overall, the evaluative 2 group appeared to be the least interested in the persuasive task”*^[5].

■ [3], [4]

Joanna L. Wolf, Effects of Annotations on Student Readers and Writers, p.23

■ [5]

Joanna L. Wolf, Effects of Annotations on Student Readers and Writers, p.25

Why are the texts from the two evaluative groups so different? Joanna L. Wolf hypothesized that the anonymity of the annotation author plays a key role in understanding that behaviour. Both evaluative groups received the same amount of positive and negative annotations. It seems that the students from both groups had a different opinion about the unknown annotator or the audience they virtually wrote for. If the student read through the text and got the impression that the annotator favored Hacks lawsuit, they might get motivated to point out their position through the essay more clearly. They might use the annotations from the annotator to find concrete arguments against those statements which could be the reason for the higher amount of argumentative 'T-Units'. It seems that the evaluative 1 group obviously disagrees with the annotator's position which would be an explanation for the different text structure in comparison to the other groups. The evaluative 2 group felt already confirmed in their attitude so they lacked motivation or engagement to make their personal position more clear. The annotator did that for them already which could create the impression that this is the obvious tenor of this text.

To verify this hypothesis that the two groups have perceived the annotator differently, 20 new students were provided with the annotated essays and they needed to fill out a questionnaire with ten statements characterizing the annotator. The 20 students gave evidence to Joanna L. Wolf's hypothesis of the different perceptions. The papers from the evaluative 1 group were more likely rated that the annotator supports Hacks lawsuit. The results showed that the students think he/she is a teacher and that he/she is biased.

How could this study help building a digital annotation system? As stated out earlier, digital data is easy to reproduce and transfer. The effect annotations have on unbiased readers are well documented here. The overall attitude towards the text seem to be unaffected by anonymous annotations. 90% of the students participating in this study had the same position on this lawsuit. Single claims were rated differently but the overall attitude was stable.

An important fact concerning the digital annotation system is the problem of the anonymous annotator. The overall attitude towards the text did not change but the way the students dealt with the text was significantly different. The personal attitude was unaffected but the attitude of the annotator made a difference how the text was

consumed. The difference between the personal and public impression of the text was influenced by the annotations in the text. Summaries resulted in less objective and factual results and more in subjective argumentative essays. This personal shift is important to know when realizing a sharing feature for a digital annotation system. An annotator profile could change the public impression or virtual audience an unbiased reader gets when reading the text. The author info should contain information which will guide the reader to the overall position of the author towards the text. It seems to be important for an unbiased reader to know the attitude of the author towards the general topic. A business consultant will probably annotate a scientific report from the WWF ('World Wildlife Fund') differently than a Greenpeace activist. This position should be reported through the author info. Suggestions could be:

- Name
- Job Position
- Favorite/recommended RSS Feeds
- Amazon wishlist (books)
- Personal iTunes Musicstore Playlist
- del.icio.us links
- flickr account

This fact is important to consider if academic, science or news texts are published with annotations. These texts are information focused and should not get misinterpreted through annotations.

08

Summary of the already discovered requirements as well as new additions based on common interaction design principles from global research.

THE REQUIREMENT LIST

08 The research so far gives a good insight into the historical development of the window, the form of annotations, the reasons why people annotate, the reading and annotation differences between paper and electronic documents, the way annotation changes if they are made public and finally how they effect unbiased readers. All these chapters ended with short analysis and mostly a requirement for the digital annotation system. To keep an overview on what the conclusions of all the research was a short summary of the founded requirements is provided here.

REQUIREMENT 001

All time present possibility to annotate content in a computer window. It is part of its appearance.

The window is the paper equivalent in the computer world. Demanding that an annotation system is working similar to paper annotation means that annotation works in every paper equivalent space. The logical conclusion is that every window which content represents something available on paper should have the possibility to get annotated. Annotations would become a system-wide feature. Since every paper-like content is presented in a computer window, the user can access the annotation possibility application independent. The annotation becomes content connected and not application connected. Meaning that the user can annotate a text for example in TextEdit and look at the same annotations in Microsoft Word later. Both are using a window both will have the annotation system build in. It is a new property for the document window which is responsible for displaying the content.

One sentence of a student in the study about personal and public annotations was remarkable because it really showed a simple fact that has often been expressed in different ways before but never to that clarity. He said in an interview: *"I don't know why I wrote some of the things I did"*^[1]. This really shows how natural it is to annotate a document. It is an unconscious act which is part of the reading experience. The awareness of the possibility of annotation is one of the key requirements to a successful annotation system. Obviously the student did not think about the annotation he/she was doing. Since it is a simple and natural part of the reading process and an always present paper feature. He/she did not need to worry if his/her pen-in-hand works on that specific paper he/she is reading or if he/she needs to convert some parts of the source document to make it work together with the pen. He/she should be able to annotate his text right were he/she reads it—in the document window.

The system wide integration also makes viewing/editing of annotation on different computers easy. No extra plugins or special annotation applications are required.

■ [1]

Catherine C. Marshall&A.J. Bernheim Brush, Exploring the Relationship between Personal and Public Annotations, p.352

REQUIREMENT 002

Simplicity. A computer annotation system needs to be fast, simple, direct and seamless.

Making choices requires attention. Providing a lot of options does not scale well with attention focused working. Software needs to show the user what it is made for. This involves not only a good presentation of the features but also communicating the limitations. The annotation system should make clear what it has been designed for. 1992 Peter Bickford wrote "*Elegant products don't do everything; but what they do, they do extremely well. What's more, they make it look easy*"^[1]. This sentence was part of a newsletter send to Apple developers regularly. Peter Bickford was chief of the human interface lab at Apple Computers and next to the quoted sentence, the text dealt with the topics of transparency (user-wise) and designing software for the mass market. He also mentions the 80/20 rule from the marketers which tells the truth in software design as well. He wrote: "*For every new feature, ask yourself whether this feature benefits one out of every five users or four out of the five. Give your priority to the latter, and if you do decide to implement features that benefit the smaller group, do it only if you can assure yourself that it won't compromise the ease with which the majority of users can get their job done*"^[2]. This puts the word simplicity into the right context. The digital annotation system should not distract the user from his main intention that is reading and working through his text. It needs to support him getting actively through it. For example selecting a highlight color should not be a choice of 16,7 millions. The requirement also mentions the word fast. Simple and fast are two properties that fit in the field of software design very well together. If the interface is simple the user will probably access the items very fast. As stated in the beginning 'Making choices requires attention'.

REQUIREMENT 003

Changing a tool should be easy and fast.

That requirement is similar to the claim for simplicity. The tools for annotating a given text will be the most frequently accessed items in a digital annotation system. Switching between them should not distract the user from the attention towards the text. The research from personal and public annotations concluded that annotations were done mostly with one tool. On the one hand this supports the hypothesis that changing tools requires attention and that the user most likely will annotate his text with one pencil. On the other hand this seems also a contradiction concerning the postulated requirement. But a pencil is impossible to translate directly into a computer tool. A pencil allows next to simple underlining also the drawing of any text decoration or notes in any size and position on the paper. In software design these tasks are divided into several tools. The requirement simply states out that switching between those tools needs to be fast and easy, to keep the attention on the text. The user should have the same comfort as if he/she would annotate a paper with a pencil.

► [1]

Peter Bickford, Interface Design: The Art of Developing Easy-to-Use Software, p. 23

► [2]

Peter Bickford, Interface Design: The Art of Developing Easy-to-Use Software, p. 25

REQUIREMENT 004

Simple Shapes, highlighting and short notes are part of basic annotation and serves most functions well.

Annotations serves different functions and each of those functions determines to some degree the visual appearance of the annotation. Serving all functions would probably end up in a complex and feature rich annotation system. But the research so far denied complexity. It is impossible to serve all functions Catherine C. Marshall mentioned in her study. There are limits to what a digital annotation system can provide without gaining too much attention to control the interface. The stated requirement points out that basic annotation possibility needs to be provided and further tool decisions needs to be balanced between simplicity and feature overload.

REQUIREMENT 005

The annotation should be distinguishable from the source.

This short sentence summarizes basically the initial inspiration for this thesis. Today's computers allow the creation of all kinds of different content with own appearances. Annotations needs to be distinguishable from the source on the first sight. There needs to be a visual separation between the original content and the added annotations. In the research the participants of the 'Reading mechanisms on paper and electronic documents' study where using bolding and italicizing to emphasize specific words. An unbiased reader could not differ between initial text formatting from the author and added annotations by the participants. A visual distinction between content and annotation is necessary to communicate the information structure clearly.

REQUIREMENT 006

Annotations need to be distinguishable by its content and author.

Annotations are due to their unconscious creation hardly categorizable. Everybody marks up a text differently. The reason behind that are found in the different functions annotation serves. Somebody highlights a paragraph to remind himself of its importance, somebody else could write a note in the margin. Paper has not only the property of showing instantly what is source content and what is annotation, it also makes clear which annotation belongs to which person. These difference are expressed through color, pens, handwriting - simply every person has its own personal annotation style. The interaction on today's computer does not leave an individual trail of the user's action behind. There is not something like a handwriting in computers. Every user uses a specific tool in the same manner. If 'User A' clicks on the italicizing button for example and 'User B' does the same, an unknown reader is not able to tell which user italicized which word. In a collaborative situation a distinction of annotations by author is absolutely necessary. The previous requirement communicates the information structure while this requirement is responsible for distinctions in the data structure.

REQUIREMENT 007

The annotation system needs to auto-save markings and notes.

In the real world every action a person does, has an immediate impact on the subject and leaves an individual 'fingerprint' behind. This fingerprint excludes the possibility for example of undo or copy actions. If somebody writes something on a sheet of paper, he/she will immediately see the result (feedback) and the just written text is automatically 'saved'.

Computer data consist out of binary code. Every action or item can be fully reproduced and undid. This nature of electronic data arises the necessity to tell the electronic machine at one point that this state of data should be saved. This saving is a conscious act and remembering it is essential.

Since annotations are right now mostly done on paper users will probably not be aware of the conscious act of saving their annotations separately. This saving would be a technology based user experience and also take the attention away from the current action - working actively through a text. Annotations needs to be automatically saved in the background to insure no data-loss and the focus on the text. Saving a document on different data levels (source + annotation) would brake the consistent file handling in current computer systems.

REQUIREMENT 008

A key feature is the support of fast and unconscious anchor-only annotation while reading.

The research unveiled that anchor-only (highlight, underline, circled words...) annotations are the most popular form of annotations. Text/Notes or emphasizing symbols like examination-marks are existing and helpful annotations but 82,2% of the annotations from Catherine C. Marshall study were anchor-only. This research shows already which focus a digital annotation system should have. These anchor-only annotations should be supported on the first hand. Keeping in mind that people are not changing tools too often while annotating should help following this requirement. Anchor-only annotations should require the less amount of mouse clicks and be the default tool when activating the annotation mode.

It is a necessity to focus on anchor-only annotations on the first hand and build the other annotation features into the system without compromising the speed of anchor-only annotations.

REQUIREMENT 009

Personal (local) and Public (non-local) annotations needs to behave and work exactly the same way. WYSIWYG. (acronym for 'What You See Is What You Get')

So far the research showed that there is a difference in public and private annotation. Annotations undergo dramatic changes before they are going to be published. The tool WebAnn used in one of the previous studies provided the user with one annotation type (compound: anchor + content) only. This was in comparison to the original paper, where the annotations had been made on, very limiting. A digital version of an annotation system should make no difference between public or private presentation from a visual point of view. The annotation the user does locally should appear in the exact same way on a different machine or in another application. The study about the International Monetary Found(IMF) showed how often annotations were used in reviewing and collaborative situations. Creating misunderstandings or confusion through displaying annotations differently would conflict with the main intention, to improve the overall efficiency in current computer document interaction.

VISIBILITY AND STATUS COMMUNICATION

08.2 An important part in the interaction design of a system or service is the topic about feedback and visibility. When a user interacts with a given system he/she reacts to a given state. This state needs be clearly communicated and displayed. Donald A. Norman states out in his book 'The Psychology of Everyday Things': "*Just the right things have to be visible: to indicate what parts operate and how, to indicate how the user is to interact with the device. Visibility indicates the mapping between intended actions and actual operation*"^[1]. This visibility will tell the user the clear state in which the system in this moments resides. Out of this observation the user can plan the following actions and give commands towards the system.

A digital annotations system makes no difference to that:

REQUIREMENT 010

It should be clearly indicated if an document has been annotated.

The presence of annotations needs be visible among the whole system. When the user receives a file wether via email or download or from a file server it is essential to see the files properties. These properties can be the size or the kind of a document as well as the presence of annotations. If a user opens a document the system needs to indicate if there is more to this document then the actual content he is looking at right now.

A missing indication would force the user to explore the presence of annotations on his own. It would not make the annotation system impossible to use but would force the user to spend time dealing with the interface instead of dealing with his goals he has towards the system he is using right now. That would be a clear indicator of bad interaction design.

But the visibility issue does not only effect documents which have already been annotated it is also an important part in the overall system design. The awareness that annotations are possible is even more important to communicate, then the overall system design or feature selection. A misconception of the annotations awareness would result in directing the attention towards the interface control and away from the text. This would not support active reading and thus be a design failure of the digital annotation system.

► [1]

Donald A. Norman, The design of everyday things, p. 8

SHARING AND FILE HANDLING

08.3 The previous research made clear how much collaboration is involved in the document-centric environments people are working in today. Exchange of information became an even more important part with the introduction of the internet. Digital data is due to its nature easy to copy and transfer. On paper, annotations are a natural part in the document handling and a digital annotation system which this thesis is aimed to design will hopefully make no difference there. If annotations will become a natural part of electronic documents a restriction to work locally only would weaken the whole annotation experience.

REQUIREMENT 011

Annotations should be shareable.

Requirement 009 demands a WYSIWYG behavior. This would only work if the target annotation system is going to have the annotation system build in as well. So shareable in the context does not just mean easy transfer of annotated document but export to different electronic formats as well.

An assumption that attached annotations would work on every existing computer would be shortsighted. There are a variety of file formats solely designed for platform independent information transfer. The PortableDocumentFormat (PDF) from Adobe Inc. is one of such formats. A digital annotation system needs to support these quasi standards to succeed in a fluid workflow. The users should not worry about technical constraints or make a decision against annotating a digital document because of compatibility issues.

REQUIREMENT 012

The file size should not be dramatically increased through the presence of annotations.

The issue of collaborative authoring is closely connected to overall file handling. File size is especially with focus of online file transfer an important issue. Small annotation files will be more likely shared than big files. That requirement results in some technical restrictions which are not part of this thesis but are reasoned in the behavior of such a system.

SPATIAL ORIENTATION

08.4 ▶ Bruce Tognazzi, founder of the Apple Human Interface Group and 14 years at Apple Computers employed, said once while talking about the Classic Mac OS Finder: “*Spatial Orientation includes not only the nesting of folders within folders, but being able to recognize a window based on its size, shape, and the layout and coloring of the objects within*”^[1]. In the field of visual communication spatial orientation has been a central aspect of guiding the viewers attention to the most informative part and communicating its intention through it. In the field of interaction design organizing information spatially is a common underlying aspect. Spatial orientation of GUI objects will not always end in a win-win situation. Spatial Interfaces are not equal to metaphors but at the end they have the same drawbacks. “[...] *a spatial interface does not seek to reproduce all the faults and constraints of real-world objects. Rather, it seeks to exhibit their advantages while transcending their limitations.*”^[2]. The research so far has been focused on studies highly related with paper from the real world. The aim of this thesis, as stated out already, is to create an annotation system which works as efficient as its paper equivalent. But it is really important to know the limits of those real world objects as well. While designing an interaction concept for something so popular as an annotation possibility it is really important to make the system not only running on a digital system but also behaving and acting like a digital equivalent to the real world. This topic is closely connected to the chapter about the implementation-centric, metaphoric and idiomatic interface paradigm introduced in the beginning of this thesis. But there is a special behavior in the real world which is normally taken for granted and thus often forgotten in the interaction design process. This behavior is normally expected by every user since that is how things behave but as mentioned already it is very often violated.

► [1]

Bruce Tognazzi, AskTog–OSX: A First Look

► [2]

John Siracusa, About the finder..., p. 2 (Spatial Interfaces)

REQUIREMENT 013

Annotations stay where the user put them.

Somebody comes into his/her office and places his/her cellular phone on the top right side of the desk for example. After some hours of working this person is going to take a lunch break. By asking this person with his/her eyes closed where would he/she expect his/her phone to be? What would be the answer? Probably on the top right side of the desk or even more clearly at the place where he/she put it in the morning. This sounds on the first hand so obvious that most people would consider it of no importance in the interaction design process. But this spatial behavior is one of the key elements often violated in the computer world. When somebody annotates a given document and he/she sends this document to his/her colleague, where is this person expecting the annotations to be? Probably there where he/she put them before.

It is the natural way things work in our real world but it is definitely not natural in the digital world. This requirement is not only a key behavior in a digital annotation system it is also the one which creates the biggest problem from a technical point of view. Anchoring annotations in a computer environment which is very modular and flexible in its appearance is a challenge. If this requirement is not fulfilled the fundamental annotations are based on-context related meta-data-is gone and the whole digital annotation system not reliable and thus useless. A computer window-the paper equivalent-is one of the most flexible interface items inside the Graphical User Interface. It can change its size, its position and easily its whole content, so all in all its whole appearance. A webpage for example: Webpages can consist out of a variety of elements (images, text, input fields, moving images, buttons, ...) and these elements might be dynamically created and positioned. Meaning that they could change size, position or even content under same URL (Uniform Resource Locator). Now a user annotates a website with dynamic content (i.e. a News-Website). Next time he/she opens up the site the user will expect to see the annotations exactly at the same place he/she placed them before independently if the window has different size or position or the content is laid out differently. Annotation is data related to a source. This relation does never break down. The positioning system needs to focus on the source content and not on the x and y position relatively to the window size. This does not only apply for anchor only annotations like highlights or underlines. The same is true for drawings and marginalia. Next to the topic of flexible positioning, there is also the issue of scalability. A website's font appearance can easily be changed through the user. Annotation needs to respond to these appearance changes. The context relation is not only a content relation it is also a visual relation. Both depend on each other to fulfill their functions.

REQUIREMENT 014

Annotations settings (last color and tool selection) are stored together with the annotated document but are limited to operate locally

But spatial orientation does not only effect the annotation itself. The basic rule that things stay where the user put them is also true for the whole interface layout/appearance. These settings needs to be document related. The research showed that different type of texts receive a different style of annotations. Marking up a news article on a website might be done differently then marking up a scientific publication on a university website. For a news site with a layout trimmed for readability (large font sizes, bigger spacing) the user might choose a highlighter to markup the document. In a scientific article with a long text underlining might be the preferred annotation tool. When the users enters the news website he should see the highlighter automatically activated and not the underliner from the last scientific annotations session. This is how spatial

orientation could improve the annotation system experience and this would make the system 'smart' and the user experience fluid and seamless.

Spatial orientation towards the tool behavior is a good example in which field a digital product could be inspired by the real world. Examining the behavior of existing products and applying them to a digital user experience might end up with a much more natural use of the system.

Informations like actual annotation tool or color choice are based on personal user preferences. They can not be generalized. This means that all this kind of informations are stored together with the annotations but is limited to work locally. If the user receives a file a colleague annotated he/she should not be prompted with the annotation tool the original author used last.

REQUIREMENT 015

Publishing annotated content should require some effort to avoid accidentally annotation sharing

There is one more thing from the real world which inspires the behavior of a digital annotation system. Again it is a subtle real world interaction but completely different to the digital world. In a long term study, Catherine C. Marshall did, she analyzed student books again. This time she interviewed the students as well and asked them about their annotations. She asked them if they do not care about publishing their sometimes quite personal annotations. They justified their action with the reason that nobody will recognize the original author of these markings. Their privacy was protect by the impact of anonymity. But what happens if this anonymity is lost? The study from personal to public annotations revealed that annotations will undergo dramatic changes if they were made public. This observation will lead to the conclusion that annotations will only be changed if the original author is going to be known. How does that related to the spatial orientation of computer interfaces? In the real world transferring documents from one place to another (handing out, selling, sending,...) is always connected with a given effort. It is something people do on purpose and they do it actively. This is different in the computer world. In the computer world the user normally browse a file in the file hierarchy and needs to open it first to see its content. Real world documents are always open and thus the annotation always present on the first sight. Thus the fact of handing out an annotated document accidentally is rare. This is different on the computer. Sending a file via email or uploading on the server does not involve opening the document. So the annotations which are related to this document are not visible on the first sight. That leads to the requirement on how to deal while transferring annotated files.

The requirement 012 demands a clear communication of the annotation status. Now the accidentally transfer of files with annotations associated sounds pretty unrealistic or the requirement 012 was not fulfilled. But this requirement only states out that it should be clearly indicated if a file has been annotated or not. Requirement 012 does not involve the exact listing of each annotation for verifying public compatibility. This requirement here wants to avoid the accidently sending of private annotations while the requirement 012 communicates overall presence of annotations.

AFFORDANCES

08.5 Next to the basic principle of spatial orientation, there is another topic borrowed from the real world, which helps interactive products to communicate their behavior more clearly.

Donald Norman built up a term called affordances which he defines as *“the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly used”*^[1].

A standard lightswitch affords to be pushed, a volume wheel on an amplifier for example affords to be turned. Alan Cooper ('father' of visual basic and proclaimer of the popular Goal-Directed™ design method persona) re-defines this quite important term by omitting the phrase 'and actual'. Taking a doorbell for example: If somebody discovers a little knob next to a door of a residence its affordance is 100% doorbell. If he/she pushes this and a trapdoor is going to open beneath him/her it does not change the fact that he/she recognized this button as a doorbell. Cooper argues that it does not matter what actually an item is going to do it is important what he/she thinks it will do. That is why Cooper omitted the phrase 'and actual' from Normans definition. But why is it that somebody recognizes this doorbell as a doorbell. On the one hand he/she has simply learned the fact throughout their socialisation. Human beings have learned about pushable objects in their daily environment and they have learned from their parents how to approach another's person home while standing on their doorsteps. On the other hand a button is triggering human bodies tool-manipulating-nature. If somebody sees something that is long and rounded for example he/she can not resist to put his/her hands around and try to move it. Or if something is standing a little bit up the surface he/she automatically tries to push it. That is what Norman calls

[manual] affordances. If something is „*clearly shaped to fit our hands or feet, we recognize that they can be directly manipulated and require no written instruction*”^[2]. This basic understanding of using something explained through its shape is a clear example of intuiting an interface.

But what does this have to do with a digital annotation system? Affordances are much better to understand than written instructions. Everybody knows the problem with doors. Going towards a door with a metal bar which has the right shape, height and position to be grasped by our hands the manual affordance says 'Pull Me'. It does not matter how often he/she walks through this door, he/she always tries to pull it and ignoring the simple sign 'PUSH'. If somebody sees round things he/she automatically tries to rotate them. No instruction necessary on how to use a steering wheel or water tap. Its affordance is obvious.

Interface designers have used this knowledge to improve the basic understanding of the computer interface. Interfaces are dominated with 3D shadings. They are not just cosmetics they show the users if they can push this item or drag it around or rotate it. These virtual affordances try to use the users tool-manipulating-brain to get things done on their computers.

The problem with virtual affordances is that it is nearly impossible to get to know the function it performs. Computer interface items are highly flexible and repetitive which makes it on the one hand understandable and usable but on the other hand it is hard to figure out its use on the first glance. There are just two ways to learn these functions—experimentation or training. The user needs somebody/something that tells him/her what this button stands for or he/she needs to figure it out on their own

■ [1]

Donald A. Norman, The design of everyday things, p.9

■ [2]

Alan Cooper&Robert Reimann, About Face 2.0 The essentials of interaction design, p.257

by simply pressing it and seeing what is going to happen. Textlabels or iconic symbols help the user figuring out the function of an item. He/she needs little hints to understand and remember its use even if the affordance tells him/her how to use it.

A digital annotation system, present in every document window, allowing a seamless paper-like annotation possibility needs to take affordances into account. It is important to orientate the visual appearance of such a system on things the user is used to. If the users are familiar with the appearance of an interface item he/she will learn much faster how to use that. If not only its appearance but also its behavior is similar to the real world equivalent the whole user experience could be more successful and the user itself will be more efficient and thus satisfied.

How this will inspire the annotation system, this thesis is seeking to present, will be showed later in the explanation of its basic appearance.

09

There are a variety of annotation solutions on the market but people still prefer to annotate on paper. Do these solutions fulfill the mentioned requirements?

REVIEWING THE CURRENT ANNOTATION SOLUTIONS

09 So far this thesis analyzed paper annotations and looked at general interface behaviors and paradigms. This analyzation resulted in a complex requirement list which will be the fundament for a digital annotation system. Before exploring the possibilities evolving out of these requirements it might be helpful looking at the existing annotation solutions on the market. Digital annotations are a quite popular demand which shows the several solutions out there in the software industry. The author of this thesis is going to pick four unique solutions to analyze their behavior in the context of the requirement list which has been constructed so far. The first annotation system is a system called MATE.

MATE

'Markup Annotator Text/Editor' (based on FreeStyle)

<p>■ VENDOR:</p> <p>Department of Computer Science University Toronto</p>	<p>■ INVENTORS:</p> <p>Gary Hardock, Gordon Kurten- bach and William Buxton</p>	<p>■ DATE OF OPERATION:</p> <p>1993</p>	<p>■ BUSINESS MODELL:</p> <p>Concept/Study and Prototype</p>
---	---	---	--

09.2 Mate has three basic modes of operation—'edit mode', 'annotation mode' and 'incorporation mode'. The edit mode (Fig 9.2.1) is simply a standard text processing application with editing features like delete, move and insert.

If somebody is emailing a MATE document and the receiver reviews the original document, MATE will switch into the 'annotation mode' (Fig 9.2.2). In this mode the user can add markings to the document like he/she would do on paper. These markings are not interpreted as commands.

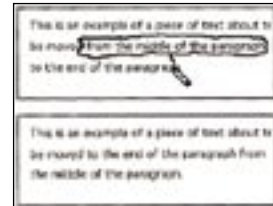
Sending this annotated material back to the original author would switch MATE into the 'incorporation mode' (Fig 9.2.3). The editor will change the interface to a split screen where the user can set which annotation should be applied to the original document. Annotations which have been executed appear in a thinner and greyed out way which results in a smart history feature to follow the annotation made so far.

If a mark appears to be thin and grey the user knows that this annotation has been applied already. Clicking on such a thin and greyed out annotation again will undo the applied annotations.

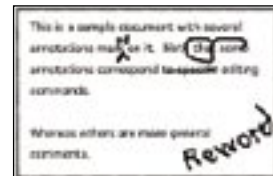
The MATE editor is a good approach for a digital annotation solution since it focuses completely on the user and is aware of the unconscious act of annotation. The user is not interfered with rules defined by technology issues. This MATE editor is simple and focuses on the outcome of the user, a good approach for a userfriendly software.

But this editor is far away from being a digital annotation system to increase overall efficiency. The shapes/gestures which MATE supports needs to be learned first. The input of these specific gesture will be easy and fluent with a stylus like input device but performing exact drawings with a mouse could be an issue which needs to be solved.

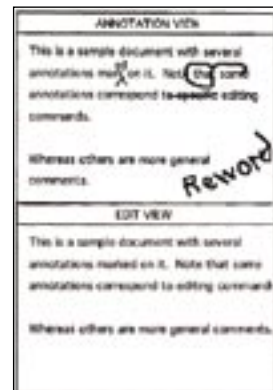
But how performs the MATE editor with a look towards the digital annotation system requirement list?



■ FIG 9.2.1 EditMode



■ FIG 9.2.2 AnnotationMode



■ FIG 9.2.3 IncorporationMode

REQUIREMENT 001

■ FAILED

MATE is a single application which uses its own spatial environment and through that its own workflow. Since the application is not available anymore the possibility of importing content from different applications is unknown. It seems that in this early prototype stage this feature has not been implemented yet and is probably never going to happen.

REQUIREMENT 002

■ ACHIEVED

This requirement is innovatively fulfilled. MATE simulates paper-like symbols and interaction and keeps the attention on the content/text. MATE is definitely simple it is also very direct and focused on the task it has been designed for. This will probably result in a fast and seamless annotation experience, even though the author of this thesis could never test the application on his own.

REQUIREMENT 003

■ ACHIEVED

Since this application has only been a prototype is not available anymore the only information source is a study from acm.org where some screenshots present the MATE interface. It seems that this application is using a one-tool interface which would be an optimal solution for keeping the attention on the source text.

REQUIREMENT 004

■ ACHIEVED

As stated out in requirement 003 already MATE seems to be a one-tool application which means that the one tool is adopting the behavior of a real pen. As mentioned in the spatial orientation chapter before "*A spatial interface does not seek to reproduce all the faults and constraints of real-world objects. Rather, it seek to exhibit their advantage while transcending their limitations*"^[1]. Copying the behavior of a real life pen is on the first a promising approach but could also lead to some drawbacks in the daily use of it. Simulating a pen with a mouse like input device has been proofed to be a problematic input form. Taking the requirement literally would result in a clear achievement. But this simple solution solves the issue with the creation of new ones.

REQUIREMENT 005

■ ACHIEVED

Since the free form ink shapes the user creates with a pen like input tool are quite different to the standard computer input it is safe to guess that annotations will be easily recognized over the source content.

■ [1]

John Siracusa, About the finder..., p. 2 (Spatial Interfaces)

REQUIREMENT 006

■ ACHIEVED

The available text sources from MATE do not mention anything towards this requirement. MATE supports three modes and they were designed with the collaborative authoring in mind. That allows the conclusion that the interaction designers of MATE had something like a distinction in mind. The screenshots from the study reveal that it seems that a separation is solved through a color coded scheme. Every single annotator is annotating over the content of the previous annotator with a different color. On the first look it seems the obvious step since this is the common way distinction is handled on paper. Color coded feedback works well on a small amount of colors but if there will be more than around six review steps this system will end in a complex annotation history. The screenshots do not make clear if it is possible to disable some colors to focus only on the latest feedback.

REQUIREMENT 007

There are no notes in the study analyzes about the saving mechanism and the screenshots do not reveal any information either.

REQUIREMENT 008

■ FAILED

MATE adopts the behavior of a free form ink pen. It allows the user to draw all kind of shapes and provides the user with nearly endless shape possibilities. Offering a wide variety of possibilities results in less focus for each single possibility. If the active user uses a stylus like input device MATE could offer a similar experience as on paper and thus support fast and unconscious anchor only annotations. Since the majority of computer system are equipped with mouse as their common input device the creation of anchor-only annotations will be difficult.

REQUIREMENT 009

■ ACHIEVED

Since MATE is the key application for this annotation system its basic software architecture provides this requirement. But it also requires from all collaborators to depend on the MATE application.

REQUIREMENT 010

■ ACHIEVED

MATE is an own application with probably an own file format. The documentation has no information about the saving process. This own file format would already communicate the presence of annotations. The documentation does not mention the possibility of hiding annotations. That means that annotations are always visible which then will fulfill this requirement.

REQUIREMENT 011

■ ACHIEVED

Since the main focus from the authors of MATE, designing an annotation environment for collaborative writing and annotations, this requirement is already fulfilled by the initial concept. It requires the MATE application for all contributors but the files itself are easy shareable.

REQUIREMENT 012

■ FAILED

The annotations in MATE are from a technical point of view just a layer on the existing text. Even though MATE is capable of interpreting free form ink from the separate layer these annotations still remain simple pixel graphics stored in a bitmapped style. Having a document with a lot of annotations added to could increase the file size dramatically. MATEs basic architecture does not fulfill this requirement.

REQUIREMENT 013

■ ACHIEVED

The screenshots provided in the acm study does not reveal any information about MATEs behavior towards scaling the main application window. Since the annotations seemed to be pixel based, scalability would be very limited. It seems that this application is quite static in its appearance. If only proportional scaling like zoom in/out is possible it is save to assume that the text will not be rearranged and the need for reflowing the annotations is not given. Annotations will thus stay were they have been put.

REQUIREMENT 014

■ ACHIEVED

MATE is a single tool application and color choices are made by the system for distinguishing between different annotation authors. Thus there is no necessity for storing any workspace specific informations.

REQUIREMENT 015

The documentation for MATE does not mention the publishing workflow while sharing annotated files.

MATE is a very user-centered approach for a digital annotation system. Its basic architecture is very much inspired by copying the behavior of real world annotation. But with this approach it also copies all disadvantages. Generally spoken MATE is a very interesting application but overall too limited to burry the responsibility for a digital annotation system.

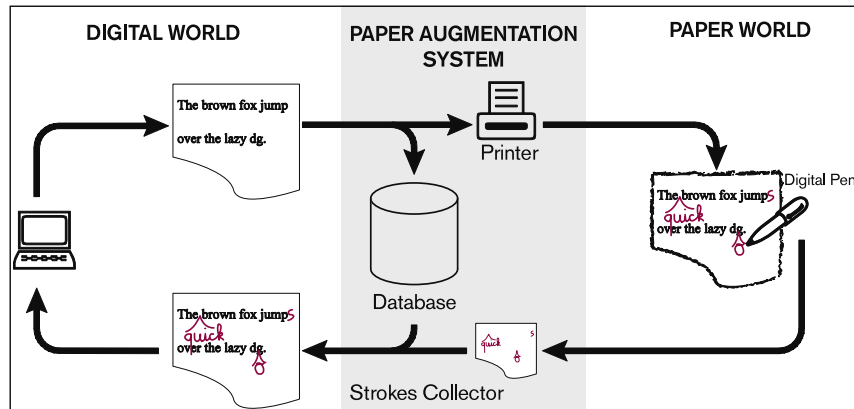
PADD

'Paper Augmented Digital Documents'

■ VENDOR: Department of Computer Science, University of Maryland	■ INVENTORS: François Guimbretière	■ DATE OF OPERATION: 2002/03	■ BUSINESS MODELL: Concept/Study and Prototype
--	--	--	--

09.3 "PADD are primarily digital documents: they are stored in a digital format, are edited using computers, and are easy to duplicate, transmit, or archive. They can also be printed on paper when the affordances of paper are needed for any particular task"^[1].

The basic concept of PADD is very different to the existing digital annotation systems which are based on the idea of annotating digital documents digitally in a computer. PADD is different since it only works with a digital pen on real world printouts. The user prints the digital document with his standard printer which adds a nearly invisible pattern to it. Now he/she can use a Logitech IO Pen to annotate as usual. The pen will scan the users activities and later transmit the data digitally to the PADD database.



■ FIG 9.3.1 The PADD Infrastructure

The PADD (PaperAugmentedDataBase) is the connector between the real and the digital world. It stores the sheet IDs for the printed pages and keeps track on the versioning of documents.

■ [1]

François Guimbretière, Paper Augmented Digital Documents, p.2

The PAP (PaperAugmentedPrinter) is from a users perspective a standard printer but adds on top of each printout a specific nearly invisible pattern to allow the digital pen storing and positioning of annotation. The current version of PADD uses an 'Anoto' pattern.

The digital pen is used to record marks made on the printouts procued by the PAP. Every sheet of paper which has an ID and an 'Anoto' pattern can be annotated.

The PASC (PaperAugmentationStrokesCollector) gets the information from the pen and request the specific document from the PADB.

The prototype of PADD was realized as a PlugIn for the Adobe Acrobat system which will be observed later in this chapter.

PADD works solely on real world paper. The requirement list in this thesis is created throughout observations from annotations on real paper. Most of the requirements will be through its basic architecture automatically achieved. Is this the digital annotation system this thesis is looking for?

The current workflow for annotating looks simplified like this:



■ FIG 9.3.2 Annotation workflow

The inspiration for this thesis to improve the overall efficiency in the document centric computer world resulted from the presence that digital data needs to get printed before the user can annotate it with a pencil. After all, these informations needs to get somehow back into the computer again. The presence of a computer in the beginning and a computer in the end of the action chain leaves inspiration for creating an annotation system solely working inside a computer to eliminate the printout and the annotation on real world paper.

But that is the basic architecture from PADD. The digital pen only works on special print outs from the digital document. PADD makes step three to four easy since it already captures the annotators movements digitally. So step four is a simple transfer of digital information which is nearly effortless and fast. PADD is a really good solution if the task would have been to improve the above mentioned workflow without eliminating the real world components.

PADD takes all the advantages from real world annotations but at the same time also its drawbacks. The fluidity, seamless, and attention focused annotation workflow is guaranteed through the use of a real world pen with a real world sheet of paper. But this real world sheet of paper also means no easy duplication. Paper can get easily lost or broken. In the real world the user can not easily search through the annotations he/she made. These are all advantages the digital world implies and which will improve the efficiency in the overall document handling.

The need for the LogitechIO Pen is another issue as well. The pen itself is only available in one color yet. The pen is not made for highlighting nor for sketching. Its ballpoint nature makes it good for underlining and marginalia. And generally the presence of this special pen itself is a necessity.

PADD is an interesting annotation system with an obviously different approach. This system could be superior in areas where annotating on a computer screen would be awkward. Sketching on architect-plans is one example. Or long texts and books. Reading online is limited to a specific length of text. The research showed that the students did most of their annotations on paper. This could have the reason that no appropriate digital annotation system was present but also that plain reading of text on screen is very uncomfortable. In this area PADD could be a handy tool since it captures the users activity digitally and eliminates the need to rework through the annotations to get them finally into the computer.

Generally PADDs concept is very interesting and promising. The key to its success lies in the integration of the digital pen into the current computer environments as well as in the acceptance of the special PADD paper. If books in bookstores would add the 'Anoto' pattern inside their books PADD could improve the workflow of books and digital documents. But as long as everybody needs to reproduce the real world text to fit PADDs pattern requirement the whole workflow is a little bit lagging.

XLIBRIS

'An Active Reading Machine'

<p>■ VENDOR: FX Palo Alto Laboratory Inc.</p>	<p>■ INVENTORS: Bill N. Schilit, Gene Golovchinsky, Morgan N. Price</p>	<p>■ DATE OF OPERATION: 1998</p>	<p>■ BUSINESS MODELL: Concept/Study and Prototype</p>
---	---	--------------------------------------	---

09.4 XLibris is an 'Active Reading Machine' which uses a high-resolution pen tablet display together with a paper-like user interface to support the key affordances of paper for active reading. XLibris allows the user to organize and search annotations as well as review, sort and filter clippings of their annotated text. Search results are displayed together with links to similiar documents and can thus help to organize and find informations while retaining many of the advantages of reading on paper.

The Xerox people define the term 'Active Reading' as not just reading per se but also underlining, highlighting and scribbling comments as a fundamental part of education and knowledge work. It is a skill that most literate people share.

XLibris addresses the advantages of paper in tangibility, free-form ink annotations and page orientation while the inventors are pointing out that their goal has been to improve active reading but not replacing paper. To achieve these goals XLibris uses the paper document metaphor which includes a tablet like display that emulates the appearence of a sheet of paper but differs radically from the standard WIMP (Windows, Icon, Menus, Pointers) paradigm usually found on popular computer systems. It provides turning pages(no scrolling), changing views and selecting ink as well as highlighting colors. The richness of interaction is achieved through the form factor of the device.

To transfer a document into the XLibris device users are using the standard print dialog inside their current application. This transfers the document to the XLibris display and the user can start reading and annotating.

The device is running Windows 95/NT and XLibris is implemented in C++ and the Win32 API. In order to treat scanned in paper and printed digital documents equally XLibris uses a 'image+text' file format. Text is only searchable if it has been running through an OCR mechanism before.

Does XLibris meet the requirements this thesis considers necessary for being a digital annotation system?



■ FIG 9.4.1 XLibris

REQUIREMENT 001

■ FAILED

XLibris is a tablet-like device which is working outside the standard computer window. So annotation can only act outside of the standard desktop computer. Thus the requirement of all time present annotations is not achieved.

REQUIREMENT 002

■ ACHIEVED

The prototype is not commercially available so testing the simplicity of the device and the user interface was not possible. From what the documentation so far reveals the whole implementation of the XLibris user interface is directed towards the user with focus on annotation so it seems safe to assume that a seamless integration is achieved.

REQUIREMENT 003

■ ACHIEVED

By its nature XLibris requires a special pen to work with the tablet. So changing tool is not really required. XLibris seems to have a menu at the bottom of the screen. This looks like a fast navigation tool to address the several types of annotation even though a more natural and maybe not so computerish approach would be more efficient.

REQUIREMENT 004

■ ACHIEVED

The pen itself allows free form ink annotations within XLibris. Marginal markings are possible but how these notes are added using the pen is not documented. The XLibris description does not mention a technology like handwriting recognition or a keyboard connector for the XLibris.

REQUIREMENT 005

■ ACHIEVED

The tablet has an own unique appearance and since some effort is necessary to transfer documents to XLibris basic distinction is easily achieved. The nature of free-form ink annotation is also pretty different to the standard digital document appearance.

REQUIREMENT 006

The XLibris documentation lists no information on collaborative authoring. It is not clear if XLibris has a kind of version management for their documents or if several authors per document are supported.

REQUIREMENT 007

The description of XLibris does not mention the saving workflow or how documents are transferred back into the computer.

REQUIREMENT 008

■ ACHIEVED

XLibris uses a pen like device for user input. Since anchor only annotations are common on paper and the preferred tool is a pen it is safe to assume that XLibris supports fast and unconscious anchor-only annotations.

REQUIREMENT 009

■ ACHIEVED

The XLibris tablet like display is required for viewing the annotations. Since the documentation does not mention a sharing feature it is safe to assume that the common way to view the annotations is the XLibris device itself. This will guarantee that annotations will behave the same even between different XLibris systems.

REQUIREMENT 010

XLibris supports several different view modes and even in the thumbnail mode annotations are displayed. Documents seemed to be stored on the tablet device so the user can be sure that documents on this device have probably been annotated. There is no information on how the XLibris syncs the annotated documents back onto the desktop computer. A clear conclusion if documents on the desktop computer will indicate that a file has been annotated or has been transferred to the XLibris can not be made.

REQUIREMENT 011

■ FAILED

As mentioned before the XLibris documentation provides no information on how this system integrates into a collaborative authoring environment. But since an XLibris is required to annotate a given document, the whole workflow of sharing annotations is demanding the presence of several XLibris systems. This makes easy sharing of digital annotation quite hard.

REQUIREMENT 012

■ FAILED

Since the documentation does not mention anything about the file saving mechanism no clear statement can be made. The screenshots may conclude that annotations do not need to reflow or scale and thus are implemented in a bitmapped way. This can result in much bigger file sizes and thus awkward file handling.

REQUIREMENT 013

■ ACHIEVED

XLibris is a closed system. The device dimension and the resulting screen real estate are defining the displayed area. XLibris displays one document at a time, thus there is no need of resizing documents inside this system. Documents which are longer than the possible display area are going to be separated into pages. Modification/Editing of source content and the resulting rearrangement of text is not implemented and thus the anchoring of annotation is realized in a quite simple manner. Position and size will always remain the same inside the XLibris system. Annotations will stay where you put them.

REQUIREMENT 014

This topic is not mentioned in the documentation.

REQUIREMENT 015

■ ACHIEVED

The reason behind this requirement was the difference between handing out real world documents or handing out digital documents to somebody. Since the XLibris device is a tablet-pc which behaves similar to paper the effort for publishing annotations is given by its basic affordances.

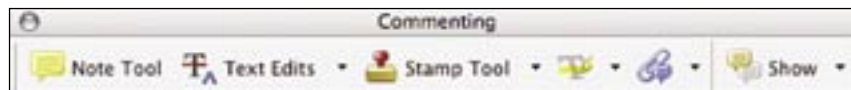
All research papers used for this thesis listed the XLibris in their references. It seems to be a popular system between interaction designer and scientists. It takes a really metaphorical approach and like the PADD system also copies the drawbacks of the paper metaphor. The necessity of owning a XLibris tablet device to annotate something will make this system hardly mass compatible. The display presence itself reduces the general unconscious act of annotations which the research considered so important. Overall the XLibris is a very interesting and somehow promising approach but the basic architecture (tablet-like display) makes it hard to use in daily environments.

ACROBAT

The PDF Flagship

<p> ■ VENDOR: Adobe Systems Inc., San Jose California </p>	<p> ■ INVENTORS: John Warnock </p>	<p> ■ DATE OF OPERATION: since 1993 </p>	<p> ■ BUSINESS MODELL: commercial product </p>
---	--	--	--

09.5 “Adobe Portable Document Format is a publicly available specification used by standards bodies around the world for more secure, reliable electronic document distribution and exchange”^[1]. Adobe PDFs preserve the look and integrity of the original documents, they are easy to share with anyone on any platform, they are searchable and they allow secure document exchange. The most common way to edit PDF files is the Adobe Acrobat application suite.



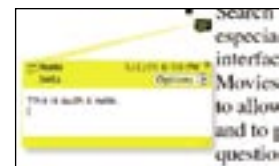
■ FIG 9.5.1 Commenting toolbar

Figure 9.5.1 shows the commenting toolbar inside Acrobat 7. There are five different categories of annotations. First there is the 'Note Tool' which could be described as marginalia, followed by the 'Text Edits' which is a specific formatting of text. Next there is the 'Stamp Tool' which inserts predefined graphics similar to real world stamps onto the pdf and this is followed by a highlighter tool and then there is the attachment tool left.

The Note Tool inserts a little Note Icon and opens a pop-up(FIG. 9.5.2) ready for typing some text into it. This is the equivalent to marginalia in real life. These note pop ups can be dragged to a suitable place.

The next item in the commenting toolbar is 'Text Edits'. By clicking on the button a pulldown menu appears(FIG. 9.5.3) All of these editing features are related to a selection of text. The interaction always requires two steps/clicks: The user selects a portion of text and then assigns an appearance to it.

The 'Stamp Tool' in the toolbar is representing small vector graphics which are inspired by real stamps from the office. The list contains items like 'Approved by' or 'Draft' or 'Sign here' and so on. The graphics are scaleable vectors and can be positioned



■ FIG 9.5.2 Note inside a PDF



■ FIG 9.5.3 TextEdits menu

■ [1]

Adobe Inc, What is PDF

anywhere in the document.

The icon next to the 'Stamp Tool' is the highlighting menu which opens a pulldown (FIG 9.5.4) similar to the 'Text Edits'. The appearance of these three tools is inspired by real world pens. In comparison to 'Text Edits' these three tools change the appearance instantly without the need for selecting a text first. The user can choose between a highlighter, an underliner or a cross out 'pen'.

And next to the Highlighting Menu there is the 'Attach' tool (FIG 9.5.5). As the wording already suggests these tools allow the user to attach files to the given document.

Adobe Acrobat offers another Markup palette which looks like this:



FIG 9.5.6 Markups toolbar

This palette allows more complex annotation shapes. Acrobat solves the problem of drawing free form ink annotations with the common mouse like input device by providing the most popular annotation shapes as fixed presets. These vectors shapes allow scaling or rotation through the use of a bounding box.

Every annotation in Acrobat has a contextual menu where the user can set its status or reply to an annotation or look at some properties. If the user would like to change the color from an annotation he/she needs to open the properties dialog from this contextual menu. The whole annotation experience in Adobe Acrobat is computer orientated and a bit implementation-centric. For each function one button, for each module of source code one dialog.

How does Acrobat result towards the requirement list in this thesis?

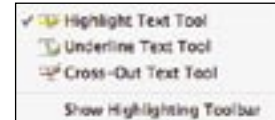


FIG 9.5.4 Highlights menu

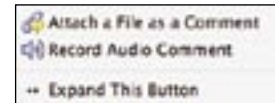


FIG 9.5.5 Attachments menu

REQUIREMENT 001

FAILED

Adobe PDF is a wide spread standard. It is a popular file format and used in complete different contexts (Web, Pre-Press, form-filling,...). But to use any of the features from Adobe PDF requires the Acrobat or a another PDF Viewer. The source document also needs to be converted into the PDF format. Acrobats annotation features are not application independent nor do they work in every computer window.

REQUIREMENT 002

■ FAILED

In which context got this requirement resolved? Exactly, in the chapter where the form and functions of annotations got explained. The requirement was requested because of the attention focus a digital annotation needs to have. Making choices requires attention. Annotating a document is an unconscious act and it should not distract the user from his/her main goal—reading/working actively through the text. Adobe Acrobat offers a variety of tools but do they make the annotation experience simple, direct or seamless? There are three different toolbars with nearly all of them resulting in sub menus. To select one item mostly requires at least two mouse-clicks. Adobe Acrobats workflow is very serial and the studies revealed that annotations are mainly done in a more parallel/multitasked workflow. In Acrobat the user has to decide which tool he/she is going to use and then look for its location in the right toolbar. This is followed by selecting the appropriate category and then select the item from the submenu. Changing the color of an annotation requires the use of a contextual menu, followed by opening a dialog box. This box provides the user with the choice of 40 different colors for the current annotation selection. There is also a button to mix an own color which results in the standard color picker window with the possibility of 16,7 million colors. This workflow is not fast nor attention focused. The interface requires quite an amount of attention to find the appropriate tools the user is looking for. It does not help to keep the user focus on the source text and all in all there seems to be too many options for too simple tasks.

REQUIREMENT 003

■ FAILED

As stated out above, three toolbars with submenus is not an easy way to change tools. There are not even shortcuts to simplify the whole annotation process. To change between underlining and highlighting always requires targeting the toolbar, selecting the category highlighting, selecting the tool/item from the submenu. This interaction does not focus on fast or easy change of tools it focuses on providing tools. This is quite a different approach then having two pencils next to a sheet of paper to work through it.

REQUIREMENT 004

■ ACHIEVED

Adobe Acrobat offers a variety of different and complex possibilities to annotate a given document. The 'Drawing Markups' (Fig 9.5.6)palette offers free-form ink presets like:



■ FIG 9.5.7 Callout tool. Cloud tool, Arrow tool

These presets fulfill different functions of general annotations. Creating those items is fairly easy but the interface is overall not very responsive. In comparison to the commenting toolbar this toolbar behaves more direct and seamless since there are no nested menus implemented.

REQUIREMENT 005

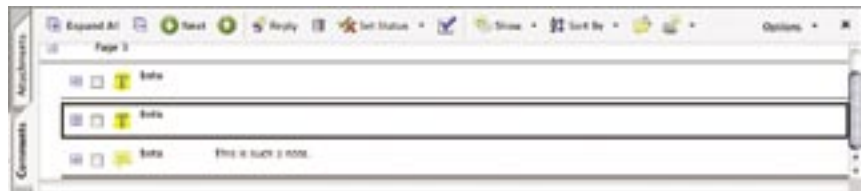
➤ ACHIEVED

The look of annotations in Adobe Acrobat is by default bright yellow. It is easy to distinguish content and annotation. The cross-out and underline tool lacks a little bit of an own appearance which 'could' result in a confusion. These two text formatting features are part of nearly every text editor and have exactly the same look as in Acrobat. It could easily be solved by providing these two tools in a more sketchy look.

REQUIREMENT 006

➤ ACHIEVED

This feature is handled computerish but it is existing. At the bottom of the active document window the user can see a hierarchal overview over all annotations. The author of the annotation is displayed right next to the kind icon.



➤ FIG 9.5.8 Authorlist

The annotation in the document itself does not show the author on the first look. Hovering over the annotation displays a little tooltip showing the author of this annotation.

REQUIREMENT 007

➤ FAILED

Surprisingly Adobe Acrobat does not auto-save the annotations made by the user. Adobe obviously understands annotations as an extra layer of content. After editing/annotating a PDF the user needs to save the pdf again. This is far away from the unconscious act of annotation the studies have revealed.

REQUIREMENT 008

➤ ACHIEVED

If the user clicks on the highlighting menu in the commenting toolbar the cursor will change from a simple text-selection tool into a highlighter-tool. This is definitely the most fluid annotation implementation in the Adobe Acrobat application. It works similar to a highlighter on paper and should support the unconscious act which is so important for the whole annotation experience.

REQUIREMENT 009

■ ACHIEVED

Adobe Acrobat is the host application which is the only possibility to create these annotations in. The PDF is capable of handling different layers of content and graphic formats. It allows the implementation of vector and bitmapped data into one PDF. It seems that the annotations are from a technical point of view just another layer of vector graphics inside the PDF. The popular PDF viewers for Mac OS X are all capable of displaying the annotations made in a PDF. This system would ensure that annotations will keep their appearance.

REQUIREMENT 010

■ ACHIEVED

When opening an annotated document the user instantly see the annotations since they are, from a technical point of view, just vector graphics added to the actual content. The hierarchy annotations pane in the bottom of an annotated document provides the user with the option to hide all annotations but by default all annotations are visible.

An indication inside the file manager to see the annotated status before opening a PDF is not possible.

REQUIREMENT 011

■ ACHIEVED

Adobe Acrobat is a wide spread format and allows several datatypes to be summarized into one file. Annotations are automatically integrated and anchored inside the pdf which results in an easy file handling of annotated documents.

The annotation system in Adobe Acrobat does not differ between public and private annotations. All annotations inside a pdf are automatically displayed by default. The research showed that annotations will undergo some changes if they are made for the public. Two conclusions are possible.

- a) Adobe Acrobat is not suitable for private annotations and therefor all annotations are automatically made with publishing in mind.
- b) Adobe Acrobat is inspired by the way annotation works on paper but simply forgot about the different handling of documents. In a digital system files need to be opened first before the content can be examined. This behavior could result in sending accidentally a pdf with private annotations attached to. That is why publishing private annotations should require some effort which Adobe Acrobat does not require.

REQUIREMENT 012

■ ACHIEVED

As stated out above annotations in Adobe Acrobat seem to be special tagged vector graphics inside the pdf. That seems to be the most promising solution to keep the file size as small as possible.

REQUIREMENT 013

■ ACHIEVED

The basic architecture on how PDF works do not allow the reflowing of text or graphics. A PDF can be scaled but not rearranged. If the user scales the window, the page displaying the content is scaling proportionally as well. This architecture easily fulfills the requirement of keeping annotations where they have been created.

REQUIREMENT 014

■ ACHIEVED

In Adobe Acrobat these settings are not stored inside the document. They are stored application dependent. If a user opens up a document all tools stay where they were placed last time he/she used Adobe Acrobat. This is a global behaviour of all editing applications which makes sense if the concept of annotating is categorized to be an editing process. That is a conceptual decision a software vendor has to make.

The research showed that annotations could also be considered as part of the reading process which would result in storing the last tool selection on a document basis and not application basis. But the basic requirement that the workspace settings are saved is achieved.

REQUIREMENT 015

■ FAILED

Acrobat does not provide a system to differ between personal and public annotations. As stated out above Adobe understands annotations as part of the content. When a PDF is exchanged the annotations will always be inside of it. This could result in accidentally sharing personal annotations.

Adobe Acrobat is probably the most advanced system for annotations on the market. The user can create a variety of different annotations, sort them and even search through them. With the wide spread PDF Format in the background it looks like a promising approach for a digital annotation system. But Adobe Acrobats implementation of some features is based on the implementation-centric paradigm. At some point the whole user attention is driven away from the text towards the control of the interface. A good example for this is the way on how the user has to change the color of an annotation.

It seems that Adobe Acrobat understands the annotation process differently then the research has revealed so far. Their concept is more in providing a rich set of features than focussing on a seamless and fluid workflow while reading.

10

First approach of designing a digital annotation system based on the research

and the requirement list. Introducing a new window property—the layer.

NEON 1.0

THE LAYER

10 The requirement list for a digital annotation system has been revealed through the research of annotations on paper and through some general interaction design principles. Now this requirement list has been compared to some of the existing annotation systems. Interestingly the first and probably most important requirement about the awareness of annotations independently from the actual application has not been achieved by any of the examples.

This first requirement is the fundament for a paper-like annotation experience which has been observed in the research. The annotation process is done **while** reading and is mostly considered as an unconscious act. The attention is centered towards the text and the annotating is just a subtle item in the tasklist of active reading.

The question that arises is how could that be achieved on a computer? In the chapter about the history of the window management, it has been made clear how the window metaphor in the users computers is connected to the paper in their real world. This connection is expressed through its native behaviour and its visual appearance (i.e. icons, wording). This thesis concluded that it is the logical path to start with a digital annotation system right where the user gets the impression of a paper equivalent. Windows are all-time present because they are the ones which are displaying the users desired content. So it is only logical to think of an digital annotation system as part of the standard document window.

A document window is a window that displays the content of a document. It is owned and controlled by an application inside the operating system. An application can open more than one document at the same time so there can be multiple instances of a document window within one application. This said, how is it possible to add to different applications with different layouts and different functions and tools an application independent feature like the possibility to add annotations to the opened doc-

ument? The question is not aimed towards the technical problems this implementation creates (which is important to ask as well), this question is directed with regard to the visual appearance. It is about the communication with the user. How can he/she recognize the presence of an annotation possibility and how can this possibility be displayed consistent throughout all applications? The content of the document window will not be able to be modified since this is owned by the application which owns this document window—the content itself should not be touched.

The answer to these questions was found by an inspiration in the real world. The question that arises the solution was looking for content which is not allowed to be touched in the real world. Visiting a kindergarten was revealing the optimal metaphor for a lot of problems. The kids in the kindergarten had a book which they were not allowed to draw in (the applications running inside the computer). But they wanted to take some of the drawings out of the book to add them to their own scenery and build their stories around it (the users annotations). They placed the book with the drawings on a special table with some sort of backlight. Then they took their sheet of paper and traced the drawings from the book. This technique allowed them to keep the original drawings inside the book untouched but work with the drawings on a separate layer. This was somehow similar to the problem described here. But this solution created a copy of the original content to work with. While inspecting the general layer and tracing idea in more detail similarities to an old technology called 'Overhead Projector' unveiled. The metaphor became even clearer. The overhead projector works with transparency foils. When for example a teacher places a foil with some content (diagram, text excerpts, ...) onto the overhead projector this content gets projected onto a wall. Since the teacher does not want to touch the original foil to use it again for later lessons, he/she adds another foil onto the original content

and add their annotations onto this empty foil. From the projection appearance this made no difference but from a data structure point of view this was the answer to a lot of questions for a digital annotation system. Adding a new layer of content by simply putting a transparency foil over the actual document window would allow the user to work with the source content without touching it. It is simply a new property for the document window.

Right now a document window is a two dimensional interface item. Its 'x' and 'y' axis defines the visible content area. If a document would receive a 'z' property a window could consist of an infinity amount of transparency foils all laying over each other onto the source content. That would change the standard document window from a flat two dimensional content container into a three dimensional stack of informations. This mechanism is not simply a new data model for a standard document window, it is also a perfect visual representation for the user.

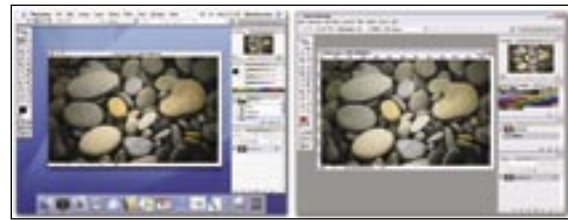
These transparency foils have an **author/owner**, they represent an own **private space**, foils clearly communicate the **distinction between author and content** and they have the **same behaviour** as paper only in a transparent way.

The chapter about affordances explained how important the communication and recognition of objects is. To profit from the just mentioned properties of a transparency foil, it is very important to clearly communicate this metaphor. If the user does not recognize the foil as a foil the properties of distinction and private space are gone. The way this metaphor is visualized does not only follow form aspects it also needs to communicate its functions. The demanded simplicity and seamless integration would not allow a visualization of a 3-dimensional stack of paper and foils. The paper/foil fundament should be communicated through its behavior. This is where the affordances of paper get important. If the user activates the annotation mode of the document window the foil should fold out of the windows title bar like a paper comes out of the printer or a letter falls into a mailbox. This animation needs to be fast and non distracting. It only serves the communication of the fact of being a transparency foil which is laying out over the actual content. This simple animation has its reason not in decoration. It is crucial to the whole communication of the annotation systems behavior. It communicates that no content gets changed only information gets added. The nature of annotations.



■ FIGURE 10.1.1 The foil animation

As stated above one important property of a piece of paper or foil is its **own private space**. Space in a computer is a really important issue. Every task the user does needs to take place in some kind of spatial environment. In the current computer systems this spatial environment is created and communicated through windows. Applications in Microsoft Windows create a non transparent, solely workspace, while on a Apple Macintosh applications are always part of the whole desktop experience. In Mac OS X each application shows the desktop or all other application in the background by default. The users workspace is the whole desktop of the computer while on Microsoft Windows the users workspace is visually limited to one application at a time.



■ FIGURE 10.1.2 Apple Mac OS X vs. Microsoft Windows

This thesis aim is not about discussing the advantages and disadvantages of both systems but it is noteworthy that these workspaces are an important issue in the computer world. Interface Guidelines should guarantee that those workspaces behave in the same manner throughout the different applications inside a computer system. This ensures a fluid user experience between different applications. The foil metaphor creates a complete own workspace inside the already present ones. The animation while turning the document window in the annotation mode communicates that the following changes are non destructive. The user gets his/her own space besides his active application to work in. He/she can do anything he/she wants without effecting the valuable content under the foil. The metaphor of private space also communicates that the foil can be deleted without changing the actual document. This behavior is common in the real world. The connection between source content and transparency foil can easily be disconnected

without changing or even destroying the source content. This essential behavior is communicated through the basic metaphor visualized through the short animation in the beginning.

Another quite important behavior of this private space is that it is—like the name already implies—private. If a document gets annotated it simply means that a foil gets attached to it. If the user would like to send the valuable source document to somebody without the personal annotations the author made to it, the annotation foil can easily be detached. This is not a feature its more like the basic nature of this metaphor. But if the original author would like to send the document together with the annotation foil he needs to declare the foil as 'Public'. The user needs to put an effort into the process of making his annotation public to avoid the accidentally publishing. The requirement 015 in the chapter about 'Spatial Orientation' demands this effort since the way digital documents are handled is different to their real world counterparts. The foil metaphor easily helps here to communicate this behavior.

The unique private space also communicates consistency. Its application independent appearance guarantees that the users instantly know how to annotate something in that application. There is no need to adjust him-/herself to the features of that specific application. Adobe Acrobats annotation system works completely different then for example MATEs annotation system. If the user needs to adjust his annotation practices depending on the application he/she is using right now, the attention gets again driven away from the text towards the interface. The application independent visual aspect of the foil sliding out of each documents window title bar will create a consistent appearance. This will result in a consistent behavior and this helps the user to focus on the annotation activity itself.

As stated out above already the foil metaphor easily communicates the modular detachability. The research resulted in two requirements which demand that the content and the annotations needs to be distinguishable from the source and even distinguishable between the author of it. This achieves the metaphor through its affordances and not by a technical implemented feature. Every author will have its own foil and every foil will be a new layer onto the document window. Thus the distinction is realized by the basic appearance and there is no necessity for displaying annotations in a hierarchical list for sorting annotations by author or toggling the visibility through a contextual menu. Its metaphor displayed through a little

animation serves all these demanded requirements.

The foil does not only help to create a private workspace or communicating its modularity it also explains the general role annotations have from a content structural point of view. By taking a look on how annotations are integrated in the most popular annotation system—Adobe Acrobat—it is apparent that annotations are equated to the actual source content. When a PDF document is opened, the annotations will instantly appear as if they were part of the original document. If the user edits the annotations he/she needs to save the document manually as if the content of the document was edited. In Adobe Acrobat the interface does not differ between the content level and the annotation level. Both is handled as simple data. This is different to paper or even to the foil metaphor. If the user opens a document which has some foils attached to he/she will not see those ones by default. There are two different content levels. The most important one is the source content. This is mostly the reason why the document gets opened on the first hand. The second level are the annotation foils. These foils alone make barely sense. Annotations are just an add-on to the source document or as John Siracusa wrote: "*Metadata is simply 'data about data'*"^[1]. These foils only work together with the source content. This is an essential content structure which needs to be communicated through the interface. Annotations alone are not new content.

By putting the source content into the first row another problem is automatically solved. The research showed that single annotations can effect the unbiased reader. If the annotations are part of the content (for example in Adobe Acrobat) he/she needs to disable the annotations first before reading the text. That is different in the foil approach. Since the foil is not visible by default the reader can first read through the text without the add-on annotations and then let the foil slide onto the document to see what a colleague has to say for example.

Current computer systems provides a nearly infinity amount of applications for different purposes or even for the same purpose. There are a lot of applications out there to write a text in. People uses these different applications because they suit their needs. An annotation system should not force the user to change his favored

■ [1]

John Siracusa, Metadata, the Mac, and you, p. 1 (Introduction)

application because it does not provide the annotation feature. The foil metaphor helps here as well. If user A likes Microsoft Word to write his text in and user B prefers Apple Textedit, then the foil paradigm let user A easily annotate his text within the Microsoft Word environment while user B views them in Apple Textedit. This is realized through its application independent appearance. Annotations belong to content not to applications.

THE TOOLBAR

10.2 The previous chapter was about the metaphor for an all time present annotation system. It resulted in using a transparency foil, originated from the overhead projectors, for displaying the annotations onto each document window. Since it is now clear on how the annotations will be displayed, the question that arises is how are these annotations created?

The research before revealed that anchor only annotations like underlining or highlighting were the most common type of annotations. 89% of the annotations were done this way and the requirement list stated out that this should be the main focus of a digital annotation system. The list also defines that note taking and simple shapes should be possible too and that the complexity and possibilities needs to be weighted against usability and fast unconscious access to these tools. The unconscious act of annotating is more important than the variety of sketching possibilities. Thus cutting down interaction in the field of possibilities makes much more sense than cutting down the field of unconscious interaction with the annotation system. The digital annotation system should provide the user with a kind of highlighter tool, a text/note tool and a simple shape tool. Those three tools should have the property of an own color which should not be a choice of 16,7 millions.

The research postulated that access to the annotations tools and color choices should be fast and seamless. From an interface design point of view there are several displaying options—palettes, toolbars, menubars, contextual menus, ...

In the book 'The Human Interface' Jef Raskin wrote: *"If a feature is not visible, we say that it is invisible. For an interface to work well, as Donald A. Norman states in his classic Book The Psychology of Everyday Things 'just the right things have to be visible: to indicate what parts operate and how, to indicate how the user is to interact with the device. Visibility indicates the mapping between intended actions and actual operations.' If an*

interface forces you to memorize the fact that a feature exists, that feature is invisible"^[1]. This results in the fact that the three tools and the color control should be always visible since they are the main interaction and can be used at any time.

As it has been stated out earlier the window is one of the most flexible interface items in the current 'Graphical User Interface'. It can change its size and position and in case of being in annotation mode the size of the foil as well. This makes a consistent positioning of the necessary tools challenging.

'Menubars' or 'contextual menus' have the disadvantage of hiding the tools on the first sight. Jef Raskin wrote that if a feature is not visible for the user it is invisible. It takes attention to make it visible and this attention should not be wasted for controlling the user interface. Using hidden controls is not an option to place commonly accessed interface items.

The popular approach for placing frequently accessed items is the 'toolbar'. Users are used to its functionality since nearly every application uses it. A toolbar can be attached to a window or to the upper menubar. If the annotation toolbar would be attached to the documents window title bar it could be guaranteed to provide the user with a consistent application independent appearance. The toolbar provides fast, visible and immediate functionality for the digital annotation system.



■ FIGURE 10.2.1 Annotation toolbar in the document window

■ [1]

Jef Raskin, The Human Interface, p.60

Communication of the tools functionality is done through an iconic representation of the action performed when the tool is activated. Additional help is guaranteed through the presence of tooltips. If a user hovers over an interface item in a toolbar a tooltip with a short explanation of the items action will appear. Besides the tooltip there will be no text indicating a buttons purpose thus its icon needs to be self-explanatory without violating overall system appearance guidelines.

All of the research and basic concepts so far have been platform independent and that is in line with the general idea of an all-time present digital annotation system. But the basic concept of using a foil for creating a new data structure and appearance to place annotations onto each document window demands some technical requirements. Native transparency throughout the whole operating system is essential for displaying the transparency foils. Right now Apples Mac OS X is the only operating system which provides native transparency. That is the reason why the following sketches and markups are all based on the Mac OS X GUI.

Apples Human Interface Guidelines include some tips and rules on how to create an icon for the Mac OS GUI called Aqua. The icon chapter of the Apple guidelines deals with application icons, document icons, icons for plug-ins, hardware and removable media Icons and toolbar Icons. Apple states out: *“The concept behind toolbars is that they provide access to items as if they were sitting on a shelf in front of you. Toolbars should conserve screen real estate while still being inviting and easily clickable”*^[1]. Toolbar icons should look like the user holds them right in front of them in his/her hand.

Selector

The selection tool on the top left corner of the toolbar is the common mouse cursor from Mac OS X. This tools intention is simply to select an annotation.

Selection is one of the basics the user does with a mouse. He/she clicks on objects and then choose something to do with it. There are various types of selections in the current graphical user interfaces. Nearly every command has an operation and one or more operands. Alan Cooper states out in his book 'About Face 2.0': *“The operation describes what action will occur, and the*

operands are the things acted upon by the operation. Operation and operand are programmers' terms; interface designers prefer to borrow linguistic terminology, referring to the operation as the verb, and to the operand as the object”^[2]. In this digital annotation system the **object-verb** order will be used. This order is the one where the user does not need to worry about the termination. In a **verb-object** environment the user gets easily confused since every operation(verb) seems possible because the computer does not know about the following operand(object). Another advantage of the object-verb order is that the user can easily execute a series of verbs to the same previous selected objects.

This basic interaction has exposed many different variants. The most often used types are discrete and contiguous selection. If data is represented as a distinct visual object which can be manipulated independently of other objects (for examples icons on a desktop or graphic objects in drawing applications) they are called **discrete data** and the selection of them is called **discrete selection**. If an application represents data in some kind of matrix, like text in a word processor for example or cells in a spreadsheet application the data is called **contiguous data** and their selection **contiguous selection**. A typical behavior for these selection types is that if a new selection is made the old selection is unmade. This behavior is called **mutual exclusion**. This is especially true if the selection before was from the type contiguous selection. For example if the user selects a text in a word processor. If he/she selects a new portion of text the previous selection gets automatically deselected. By looking at the research again the participants of the study 'Reading mechanisms on paper and online' criticized the behavior of the computer concerning the note taking process. They selected a word/sentence/paragraph to copy them to the note sheet. They needed to switch between the note taking sheet and the source document for every single contiguous selection because of the mutual exclusion behavior. Since moving between documents is more time consuming then moving between sheets of papers, the research conclusion pointed out to improve the copy and paste workflow concerning the digital annotation system. But the problem in the copy&paste workflow is that every selection gets deselected through another selection. The contiguous data (text) which has

■ [1]

Apple Computers Inc., Apple Developer Connection: Apple Human Interface Guidelines, p.129

■ [2]

Alan Cooper&Robert Reimann, About Face 2.0 The essentials of interaction design, p.281

been annotated with a highlighter for examples turns itself into discrete data. The annotations appear as single objects which can then be selected with a single click. If mutual exclusion would be turned off the user could sequentially click on the annotations he/she would like to copy onto his/her note sheet for example. This kind of selection is called **additive selection**. This additive selection would improve the interaction of note taking since moving between documents is reduced to a minimum. The user selects the annotation he/she would like to see on the note sheet and simply copies it over.

Besides using the additive selection feature for copy&paste only, annotations could be deleted with that type of interaction as well.

Highlighter

The second tool next to the selector is the highlighter. As the Apple Interface Guidelines states out toolbar icons should look like if the user holds the tool in his/her hand. If he/she wants to highlight a paragraph/sentence/word the icon should represent the look of a common highlighter. The most common highlighter is the Stabilo Boss pen. The first icon sketches focuses on presenting the Stabilo Boss shape in the Apple Aqua look.



■ FIGURE 10.2.2 Inspiration sheet and first icon approaches

When the Aqua-like highlighter icon was finished it turned out that its photorealistic reproduction is going to be a problem. Standard toolbar icons have the size of 32x32 pixel. Scaling the highlighter icon down to that size made it unrecognizable on the screen. The popular website xicons.com offers a free iconset^[1] which represents different tools on a desk. Their highlighter icon

■ [1]

WatiWorks Icons are Copyright © Wati Larke, 2003

worked much better in 32x32 pixel. For the first interface sketches this iconset was used.

The highlighter tool is used for the anchor-only annotations. Its main focus is the simple highlighting of words as popular on paper. The highlighter tool is working through click and dragging. Its behavior is equal to a standard contiguous selection of text. The highlighter is a direct manipulation tool which will instantly show the appropriate feedback—in this case the highlighting of a word.

Pencil

“Annotations on paper are highly individual in form: digital annotations should respect this fluidity”^[2]. The highlighter has been solely designed for creating highlights. The requirement 004 demands next to the highlighting and a note/taking solution the possibility for drawing simple free form annotation shapes as well. That is the intention of pencil tool. The pencil behaves like pencils in the real world. It allows the user to draw any kind of freeform shape he/she can think of. The pencil does not offer a selection of a brush size for example. This is reasoned in the demanded simplicity for a digital annotation system. The user should not try to draw complex things with that pencil. Its focus and attention should reside on the reading and annotation process of the text. A choice for the thickness of the pencil is not going to add that much information that would justify the complexity which is gained through the presence of these specific graphic options.

The mouse

“With a thin-bodied stylus, we can get very precise control of the point, but with the palm-sized mouse, the muscles in our fingertips don’t come into play the way they can with a pen. This is why we cannot enter handwriting practically with a mouse”^[3]. That is an example probably every computer user has tried and miserably failed. For annotation this is a crucial behavior, which needs to be taken into account. Drawing with the mouse a question mark or exclamation mark is an attention consuming interaction and will probably end in something far away from the look of a question mark. In Neon the

■ [2]

Catherine C. Marshall, Annotation: from paper books to the digital library, p.139

■ [3]

Alan Cooper&Robert Reimann, About Face 2.0 The essentials of interaction design, p.267

solution for drawing complex shapes with a mouse like input device is to exchange the drawings with a more simplified version of their appearance. The computer will then exchange the simplified version with the complex shape. This system is called **gestures**. The data which the user enters is going to change from simple drawings into computer commands. Right after the user completed one shape/command the computer will execute it and replace the drawn shape with the appropriate one. The reason why gestures are a really fast input source is that the operator and operand are entered with a single stroke and since gestures appearance is iconic they are easier to remember than textual commands.

Since gestures are drawings as well it is important to figure out which gestures could work well with a mouse input device. A user-test has been designed to understand which kind of shapes are problematic to enter. But more about that test in Appendix B. This test has never made it to the public. The reason for that will be explained in a later chapter.

T Note-taking

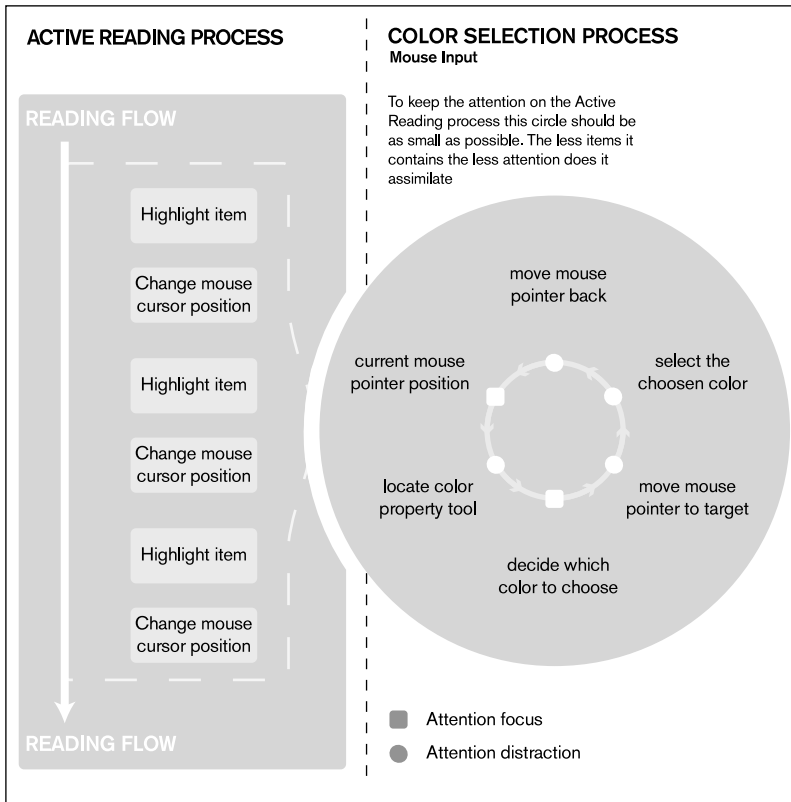
The note taking tool takes as the name already suggests notes. If the user clicks with this tool activated somewhere onto the transparency foil he/she will get a prompt to insert some text. The user does not have a choice of which font he/she would like to use or which size to take. It is limited to pure text input to keep the focus on the active reading process.

Color choices

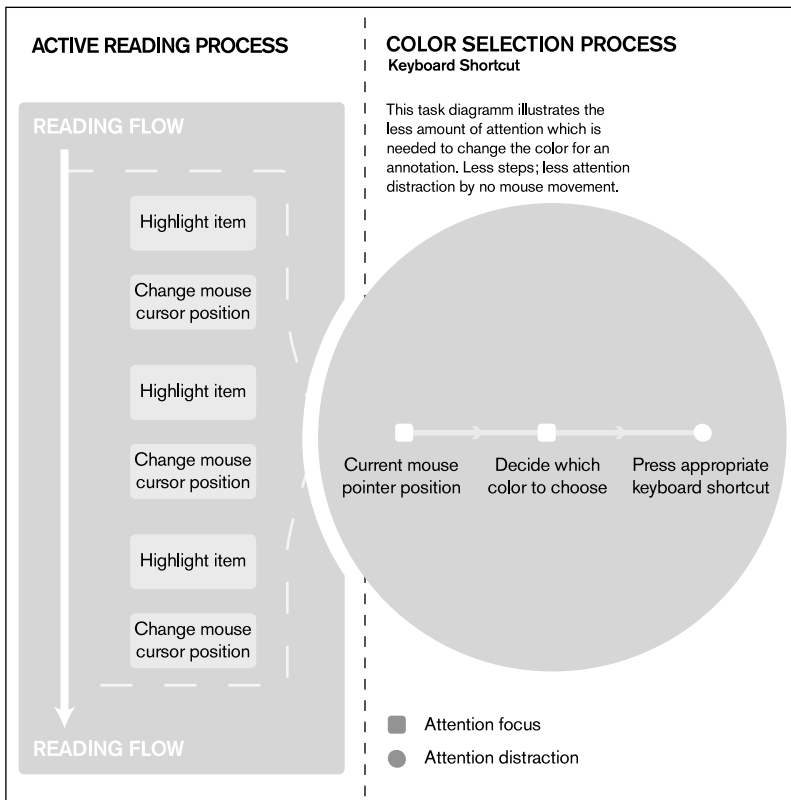
In the real world changing the color of a pen means to change the whole tool. The color is a fixed property of a pen as well as its thickness or pen-tip-shape. The computer with its object oriented interfaces and popular direct manipulation paradigm allows the user to change single properties from selected objects—in this case the highlighter, pencil and note-taking tool. If there is no need to change the tool if the color gets changed it looks like a real improvement over the annotation interaction in the real world. But the basic interaction cycle of what is necessary to change the property is surprisingly similar to the interaction of changing a tool. This is the reason why requirement 003 'Changing tools should be easy and fast.' is also valid for changing properties. The interaction cycle for changing the property color is visualized on the following page in figure 10.2.3.

Rectangular shapes stand for attention focused interaction while circled shapes display attention distracting interaction. The big rectangle on the left side of the diagram displays the standard active reading flow with annotating with a highlighter. The circle on the right shows the attention distracting interaction of choosing a different color. By examining the action cycle closer it becomes obvious that the 'current mouse pointer position' and 'decide which color to use next' are the only two items in the list which do not take the attention away from the text. All of the other items are interactions with the mouse to control the interface which could distract the user from its task of active reading. To improve this interaction it is necessary to give the option of changing the properties without using the mouse. This is done via keyboard shortcuts (for example hitting the number keys 1–6). The interaction cycle would change to figure 10.2.4.

This workflow is much more optimized for focussing on the text instead of using the limited amount of attention to control the interface. Keyboard shortcuts can always be only an extra option. An interface which demands the knowledge of invisible shortcuts is hardly to use for beginners. Shortcuts provide successful interaction improvements for advanced users but novice users will hardly use them. A shortcut is always an optional improvement but can never be a single input method.



➡ FIGURE 10.2.3
Color selection process for the mouse input.



➡ FIGURE 10.2.4
Color selection process for the keyboard input.

NOTES AND APPEARANCE

10.3 The previous chapter explained the tools Neon provides. Part of this list was the note-taking tool. It is a simple and necessary interaction inside the annotation system. There is only one major problem with the note-taking possibility: It is really hard to predict where to find white space to add the text to it. Web-site layouts are quite different if compared to the layout of text documents for example. Poster designs from Adobe InDesign are again very different in their appearance. It is important to keep the note appearance consist throughout all applications. This is not reasoned in an aesthetical point of view. If text and mark-ups are connected or related to each other, this relation needs to be made clear. In Adobe Acrobat this problem is solved by using the metaphor of stickies and sort of speechbubbles to relate the note to the mark-up. But this solution involves more pointing and clicking with the mouse. The task diagram in the choosing color section before, showed very well how much more attention a mouse interaction requires. In the sticky solution the user needs to decide about the place of the sticky and about its size. This involves mouse interaction and distracts the reading flow. Stickies are generally speaking not a bad idea but the amount of intention this interface item requires is not in line with the requirements for a digital annotation system.

The only consistent solution would be the creation of a specific white space for note creation. Apple Mac OS X provides a child window which slides out from the parent window, while the parent window is open. This child window is called drawer. Apples Human Interface Guidelines states out that: *“A drawer’s contents should be closely related to the contents of its parent window”*^[1]. A drawer is always part of the parent window. If the parent window is active the controls or items in the child

window drawer are not dimmed or deactivated. They are accessible like the controls in the parent window. The common way a drawer gets displayed is that the user activates it by clicking a button or choosing a command. When a drawer opens or closes, it appears to be sliding from behind its parent window, to the left, right, or down. To support the sliding illusion the drawer is always a little bit smaller than its parent window. The user has the possibility to resize the drawer. By dragging on the longest edge of the drawers border the user can change its width or if the drawer is positioned at the bottom its height. If the user sets a new size the drawer remembers its size the next time the document gets opened.



■ FIGURE 10.3.1 Drawer on the left side of the window

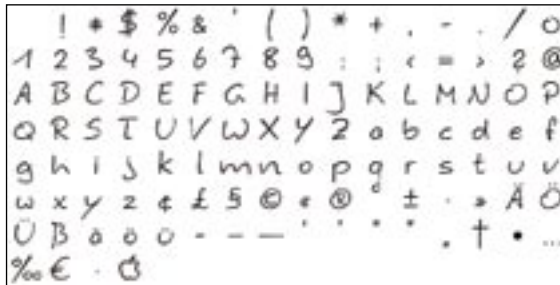
■ [1]

Apple Computers Inc., Apple Developer Connection: Apple Human Interface Guidelines, p.176

Typography

The font used for the notes in the digital annotation system is a non commercial font called Hybi. It is characterized through a quite good scalability and a spacing which is a good compromise between letters per lines and readability. The drawers space where the font will be displayed is limited. The needs for a font inside the drawer are good readability in even small sizes and a look and feel that fits into the whole annotation appearance. A handwriting font is not the most space efficient and readable solution but this font helps to communicate the aims a digital annotation system want to fulfill. As stated out earlier a good interface does not only communicate its strength it also makes very clear where its limits are. The provided note-taking space in the drawer is only intended for fast, sketchy notes. If the font would be standard grotesk or serif font from the operating system the distinction between source content and annotation is not granted anymore and the fast and sketchy character of the handwriting font is not communicated anymore. Thus to stay in line with the annotation appearance and a strong interface communication the handwriting font Hybi is the optimal solution for these interface sketches.

inside their computers. Next to the typography and the appearance of the highlights, the drawings made with the free form annotation tool will follow this theme by its basic tool design. The user is not going to draw exact straight lines or symmetric circles with the mouse. Thus the whole appearance of the user input will follow one appearance which simply communicates its purpose.



■ FIGURE 10.3.2 Fontcard of the freetype Hybi

General Appearance

The whole sketchy theme is not limited to the typography only. Again this appearance is not reasoned in a aesthetical decision. As mentioned in the introduction of this thesis, an important part in the work of an interaction designer is the design of the applications behavior. This behavior is responsible for fulfilling the users goals. The sketchy appearance of the annotations is one part in the communication of an annotation systems intention. The user should automatically get the impression of annotations on paper. This will tell the user what do with the software and how to use it in which situation. Its simplicity and user goal focused design should help to integrate the software into the different workflows users have

VISIBILITY

10.4 The menubar is one of the oldest idioms in the graphical user interface and its design one of the most unquestioned items. The menubar is placed on top of the main display (Apple Macintosh) or at the top of the main application window (Microsoft Windows). Menus inside the menubar behave like standard drop down menus. The user clicks on one item in the menubar and the menu unfolds in a panel right under the clicked menu item. Besides the one-dimensional solution there is also the possibility of two dimensional so called cascading menus. These menus unfold with an extra panel next to the last rollovered item. Cascading menus should be avoided for frequently accessed items.

By default an Apple Mac OS X menubar consist at least out of the 'Application menu', 'File', 'Edit', 'View', 'Window' and 'Help' Menu.

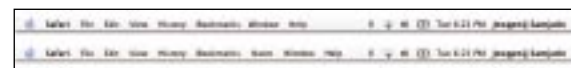
- The 'Application Menu' first purpose is to display the name of the active application inside the menubar. In the menu itself the user will find items which regard the control of the application in whole. The user can find items like 'Quit' the application or open 'Preferences' or 'Hide' the application.
- Commands in the 'File' menu applies generally to one specific document inside the active application. The user can find 'Open', 'Save', 'Print' commands inside the 'File' menu for example .
- The 'Edit' menu provides commands to change/edit the specific document open in the users application right now. Besides the changes it provides also controls for sharing data between application. This includes cut, copy, paste via the clipboard. The popular 'Undo' and 'Redo' commands are placed inside the 'Edit' menu as well.
- The 'View' menu offers a set of commands that will change the appearance on how users see a windows' content. It does not involve the management or arrangement of several windows the 'View' menu is focused on the content of the active document

window. In the Mac OS X Finder the 'View' menu is used for switching between icon and list view or for hiding/showing the toolbar.

- The 'Window' menu provides commands for organizing and managing an application's windows. It lists for example the open document windows and provides functions for minimizing or zoom.
- The last default menubar item is the 'Help' menu. It is always the rightmost menu in the menubar. As the name already suggest this menu contains information on how to troubleshoot an application.

The menubar is in its behavior and appearance application independent. The complete look&feel is provided by the operating system and guarantees a consistent user experience. Single applications will add, with regard to the mental model of the users, menus or menu-items to control the application.

If a digital annotation system claims to be application independent it is only logical that there should be a menubar item which let the user control the annotation system. From the short summary above the annotation system seems to fit into the 'View' or 'Window' menu. The problem with that solution is that it will not communicate its all time presence annotation possibility very well since the user needs to open the 'Window' or 'View' menu to finally see the annotation system controls. To follow Jef Raskins argumentation "*If an interface forces you to memorize the fact that a feature exists, that feature is invisible*"^[1]. That is one of the reasons why a digital annotation system should represent a complete own item in the menubar.

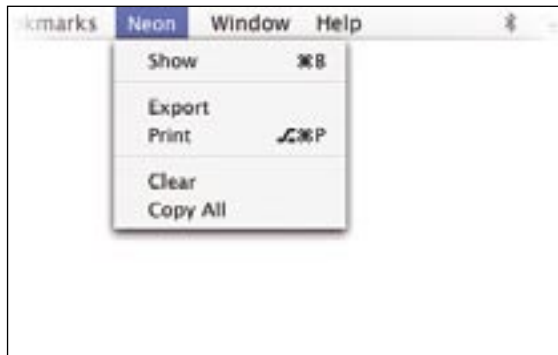


■ FIGURE 10.4.1 Old menubar vs. new menubar

■ [1]

Jef Raskin, The Human Interface, p.60

The menubar item controls features from the annotation system which are standing in concurrence to standard application features. The 'Print' menubar item in the 'File' menu is a good example for this. Nearly every application allows the user to print the current document. Apple Mac OS X offers a standard print dialog which is often used by shareware or freeware developers and there are some applications which use custom print dialogs since they offer the user more flexible options. This 'Print' command is always placed inside the 'File' menu. If the user would like to print out his/her annotated document together with the annotations he/she can not use the print command from the 'File' menu since this an application feature where the operating system can not interfere. That is why the digital annotation system needs to have own dialogs for commands which an application could already have.



■ FIGURE 10.4.2 Open menubar item

- **Show:** The first menu item toggles between hide/show. It activates the digital annotation system and starts the animation which unfolds the foil out of the title bar.
- **Export:** The requirement list concluded that sharing annotations is an essential feature. Since the annotation system will not be present on all computers at the same time there needs to be an option for exporting the current document together with the annotation to fulfill the requirement of exchange. This could be an image for example or it could also be an Adobe PDF.
- **Print:** This is the item which the user clicks on if he/she would like to print out the current document together with the annotations
- **Clear:** This will delete all annotations on the current foil.
- **Copy All:** This item copies all annotations into the clipboard.

The menubar decision is not only based on providing the right functionality. The design conclusion also involves the fact of visibility and notification. The chapter about visibility explained already the very important issue of the awareness to annotate a given document. An interface is the translator, is the communication link between the human and the machine. A clear communication about the actual status is very important. Donald A. Norman states out in his book that the visibility of items is bridging the gulf between execution and evaluation.

Gulf of Execution:

"The difference between the intentions and the allowable actions is the Gulf of Execution"^[1].

The Gulf of Execution represents the difficulty the user has in translating a psychological goal into a physical action. For example, the goal 'learn more about avocados' is purely psychological. The user of an internet browser will need to see the available links, recognize the appropriate choice, and use the mouse to point and click on a specific link. Thus, in order to learn more about avocados, the user needs to execute a set of physical steps.

Gulf of Evaluation:

"The Gulf of Evaluation reflects the amount of effort that the person must exert to interpret the physical state of the system and to determine how well the expectations and intentions have been met"^[2].

The Gulf of Evaluation represents the users difficulty in evaluating whether the response of the computer system meets the desired goal. Taking the just mentioned example, the user would need to comprehend the system response, examine the results, and evaluates whether it presented the proper information about avocados. In the same way that the user needed to translate a psychological goal into a physical action sequence to bridge the Gulf of Execution, the user needs to compare the physical response of the system against the psychological goal to bridge the Gulf of Evaluation.

Visibility is one important topic to bridge the gulf of evaluation and the gulf of execution. If the user can not see the levers and controls to fulfill his goals the execution will not be done. And if there is no feedback to the action the user did there will be no evaluation of the given action and thus the action will be claimed as failed.

Besides the presentation of controls and levers to annotate a given digital document there is also the ne-

■ [1], [2]

Donald A. Norman, The design of everyday things, p.51

cessity to communicate the actual annotation state of a given document. If the menu for controlling the digital annotation system would be nested inside a the 'View' or 'Window' menu it is difficult to clearly communicate the actual annotation state. That is much easier if the digital annotation system has an own menu point to use for notification or the control of its system. There are two main places to notify the user about the presence of annotations:

unopened file (i.e. Icon)

As mentioned earlier already file handling is very different to the paper handling in the real world. Digital files are by default in a 'closed' state. To view the content of a digital file the user needs to open it. To recognize a file as the right one the user needs clear identifiers. That could be its 'name' or 'created in June' or 'last opened yesterday' or 'has been annotated'. One of the filesystem task is to provide the user with as much information as necessary to identify a given 'closed' file. The user needs to know what he/she can expect when opening the file. In the first chapter about the history of windows the research showed that the paper inspiration did not stop by using 'windows' or the wording 'document', even the generic document icons are inspired by a sheet of paper. That makes it the obvious choice for using the document icon as an annotation notifier inside a filesystem



■ FIGURE 10.4.3 Notification on icon level

opened file inside an application

The second notification domain is the opened document. If a file has been opened already the user should still be able to see if there is more to the open document than its plain content. The layer chapter explained already the role annotations have in a digital annotation system. The main aspect and focus is directed towards the source content. Annotations are meta data—data about data.

That is why in comparison to Adobe Acrobat annotations are invisible when a user opens a document. He/she is going to look at the document as if there are no annotations attached to. The user gets notified about the presence of annotations but a digital annotation will give the user the choice to see the data about the source data or simply work through the document without looking at the annotations. Here is the visualization on how the user will be notified about the presence of annotations in an open document.



■ FIGURE 10.4.4 Notification on menubar level

11

Evaluating the research efficient user experience.
towards real user goals in Introducing Personas.
order to create the most Goal-Directed design.

GOAL-DIRECTED DESIGN™

11 ▶ Until now the design approach of the digital annotation system has been based on research which focused on paper. The solution so far centered around copying the workflow, users have while annotating text on paper. The problem with this technique is that a good workflow on paper does not automatically result in a good workflow on a computer. Each medium has its own properties and designing the behavior of a digital annotation system needs to deal with specific digital behaviors.

Besides the issue that workflows can not be easily transferred from one medium to another this thesis has not talked about the users of a digital annotation system. That is probably the weakest point on the solution so far. The current solution has some good basic ideas but to provide an efficient user experience the properties of the user needs to be known. There are several different models to analyze the users/targetgroup to design the software towards the needs of the users. The most popular method right now is called personas. Alan Cooper has been mentioned and quoted in this thesis before. His popular book 'About Face 2.0 – The essentials of interaction design' is an international bestseller but he is even more popular for introducing the Goal-Directed™ design method called **Personas**.

This method is popular for getting everybody in the development team involved into the users concerns and at the end this is the reason why the software is made. Alan Cooper is credited to be the first person to use the persona model in 1980. He is a "*software inventor, programmer, designer and theorist*"^[1] and is also called the Father of Visual Basic. In 1992 he founded his own company called Cooper Interactive to promote his promising method of personas.

Personas are "*composite archetypes*"^[2] from the actual user research and target group definitions. They stand for the real users outthere and show their intended goals with the product in a very non special manner thus all product participants (business, design, development) can understand what type of customers they are making the software for. It is not a system for usability experts or a model only understandable by technicians, it is a simple description of a real person which stands for the

needs and goals from the specific target group. Personas includes "a real name, an age, a photo, personal information (including family and home life), work environment (the tools used and the conditions worked under, rather than a job description), computer proficiency and comfort level, pet peeves and technical frustrations, attitudes, motivation or „trigger“ for using a high-tech product (not just tasks, but end results), information-seeking habits and favorite resources, personal and professional goals."^[3] These user models are not made up or are a result of the creativity from the marketing consultant, these archetypes are the result from real user interviews and user research. A persona is an example person from one group of intended customers. The key behind a persona is the goal from each archetype. These goals are divided into 'Life goals', 'Experience goals', 'End goals' and these goals define the success of the actual product. That is why personas are part of a method called **Goal-Directed Design**.

Common mistakes during the user centered design process is to stretch the users needs so it becomes an elastic user, or the team creates self-referential designs, or the software is solely designed for edge cases which rarely occur.

The 'elastic user' comes into the design process when the user has not been clearly defined in the beginning. Every member in the development team will stretch the user so it fits the personal argumentation. If for example the programmer would like to provide the user with hierarchical menus to browse for functions or necessary files he/she might change the user from a novice to a computer accommodate user to justify his solutions.

'Self-referential design' happens when the development team creates solutions which are directed towards their own needs and own goals and completely ignoring the customers point of view.

'Edge Cases' are these type of rare interaction combinations or feature request which will rarely occur or seldom be used in an applications life. If the development team focusses to much on these special situations or features the main direction of the application is lost.

Personas are a model for helping the development to prevent those situations. With personas the user has a concrete identity. This identity is used not only by the design team, it is also valid for the implementation process

■ [1]

Alan Cooper&Robert Reimann, About Face 2.0 The essentials of interaction design, p.2

■ [2]

Alan Cooper&Robert Reimann, About Face 2.0 The essentials of interaction design, p.55

■ [3]

Alison J. Head, Personas: Setting the Stage for Building Usable Information Sites

as well as for the marketing department. Personas are a communication fundament during the development process.

Here are the three Personas for the digital annotation system which should make clear if the digital annotation system is already suited for the specific users or if there is room for optimization.

JOHN



John is a 49 year old translator. He has an engineer degree in electronics and likes to travel a lot. Through this traveling John found a lot of friends all over the world and took classes in technical and electronic english. He is working as a freelancer and his customers are big international companies who needs help in translating their manuals or internal instructions very accurately. He mainly works with the Microsoft Office Suite as his customers do the same. John is a true Microsoft Word genius. All reviewing and annotating is done through that application and sometimes he is a little bit upset about the poor Word knowledge of his dialog partners. He is using an international standardized set of reviewing symbols and conventions, and dislikes it if some of his partners are violating these. He loves optimal, fast and efficient workflows. His work is dominated by accuracy and precise articulation. He often speaks with technicians himself to avoid any misunderstandings.

John works mainly digital but at the weekends he travels to his girlfriend without any computer accessible. One of his favorite hobbies is photography. His pictures are really precise and accurate and John really likes to know every technical detail of his camera. In his spare time he is quite active in a camera forum on the internet. He really likes the internet and takes online magazines and creative communities to get inspired and exchange technical tips and tricks in photography. John understands his computer mainly as a tool. If something is not really working he is getting quite fast impatient. That is also true for learning new tools. If it requires some time to succeed (i.e. Adobe Photoshop) John is going to look

for external help (Tutorials or HowTo's) from the internet or is going to ignore that application for the next two weeks. John needs fixed repetitive workflows to understand the software. It is valid to conclude that John is sort of a manual guy, abstract workflows (own scripts or actions) are not his preference. With neon he would like to optimize the overall customer communication and forces his dialog partners to use standardized correction symbols to avoid misunderstandings and increase overall efficiency. Less questions means faster results. His passion is not in the computer itself, his passion lies in the outcome. One thing that is important to keep in mind about John is that he is wearing special glasses—varifocal glasses. These glasses are quite common for older people who need to work quite an amount of time in front of the monitor. Overall they allow a better sharpness and a more natural image of the monitor but they also limit the eyes to a specific frame. John complaints that the need of moving his eyes over long distances on the screen results in tiredness and is quite distracting. That is why he is keeping his windows always pretty narrow to limit the amount of words per line and keep the eyes focused on a small area of the screen.

CHRISTOPH



Christoph is a 31 years old art director at a small but pretty popular design company delivering mostly web products. On his 'I am a good person' list there are quite some achievements—styleguide and digital Corporate Identity for a leading sport fashion company, several books and awards and last year he was part of the international Art Directors Club Award Jury. Christoph is quite happy with his current situation even though he is always looking for new challenges and items for his 'I am a good person' list. Since visual expression has never been enough for him he used to be a drummer in a small punk band. But his daily job does not allow so much spare time, so he started making electronic music with his computer.

Christoph is working with the most popular industry

leading graphic tools to get his job done and has build up quite an unique and effective workflow. He is open to new technologies and applications but if their learning curve is too steep it is most unlikely that the new stuff is going to take a place in his workflow. Christoph needs tools which he can rely on and if they do not work as promised it is quite sure that they are going to be replaced real soon. Christoph always tries to be the best in what he is doing but usually not in the way the others are trying to achieve it more in his own unique way. He is not the tech guy but he likes to play around with new technology. Software he uses needs to be flexible to fit his unique workflow.

As an art director Christoph has quite an intense contact with the customers discussing about the aims and achievements. He is the communication link between the sales&design departement and feels quite comfortable there right now. Neon should bring up new possibilities to communicate his ideas with customers and colleagues. Feedback is an important part in the daily creative business. It is the most common and probably most effective way to shape the idea/product to become perfect. If Neon could improve the transfer of feedback, it is most likely that Christoph is going to use it.

from all around the planet. Their local telephone is using the latest VoiP technology and thus Thomas whole communication went digital.

For sure Thomas owns the newest smallest digital camera on the market to make some shots in the weekend or on holiday. His favorite setting is AUTO since Thomas is not interested in photography that much. It is hard to imagine Thomas without a device that does not require electric power.

In the university the lessons are provided half in classes half online. All documents are handed out digitally and the results from each semester needs to be delivered digitally too. Thomas always tries out new applications to optimize his workflow and his operating system is always up to date.

Working with technology is sort of fun for Thomas. He loves every new application and discovers its possibilities in a very playful manner. He is very open to new workflows while innovation is the motivation to give the new software a credit. Neon could help Thomas on the one hand in his daily school life but it might also be interesting how Neon and his private blog are going to coexist. Thomas is a nice tech guy with his eyes directed towards the future.

THOMAS



Thomas is 24 years old and studies digital multimedia technology(web). He is as digital as a person can be. Thomas loves every new gadget released on the web especially from the asian market. His girlfriend is always worried about the bank account but Thomas manages it through some freelance work to keep a plus there at the end of each month. He is making websites for some companies in his town and has earned quite a good name from it. He is always using the latest technology and is quite versatile with the newest web programming trends. In the evening while watching TV with his girlfriend Thomas updates his blog with his WLAN equipped laptop. His RSS Readers counts over 250 subscribed feeds and his three chat lists are filled up with persons

PROBLEM ANALYSIS

11.2 The three personas introduced in the previous chapter are providing a concrete frame in which the digital annotation system needs to work in. By looking at the goals these three personas would like to achieve, it should be easy to analyze how the current annotation solution fulfills the job. John, the translating engineer, annotates mostly for reviewing. He does that with official standardized correction signs to avoid any misunderstandings through the interpretation of custom symbols. Christoph, the art director, mostly annotates images/layouts to review them internally in his company or to send more detailed feedback to the customer. Thomas, the student, uses annotations more in a collaborative situation together with a colleague from the university or he publishes annotations on his personal blog articles.

Johns work is characterized through annotating Microsoft Word documents. How would he create his special correction symbols with the current digital annotation solution? The only possibility is the pencil tool. This tool allows him to create free form ink shapes with his mouse. Will John favor drawing with the mouse over drawing with a real pen? How does he connect his special shapes to specific words/passages? How to keep the symbols in the right context when the document windows resized? Will the digital annotation system be more efficient in comparison to printing out the Word document and annotate it on paper?

By comparing the skills of John and Thomas to the skills of Christoph it is obvious that Johns and Thomas' mouse experience is limited to control an interface through pointing and clicking. Christoph is the only person who works creatively with the computer and might provide a much more solid virtuoso mouse handling. This skill is not necessarily present if the user has only experience in pointing and clicking inside a word processor or an internet browser. But precise mouse control is required if John would like to draw a question mark in the marginalia or if Thomas would like to insert an asterisk inside

his blog text. To even allow John and Thomas the input of complex shapes the current solution use gestures in combination with the pencil tool for complex shapes input. Does this help John and Thomas to get thier work done more efficiently?

Basically there are five issues that might happen if gestures become the main control element inside an interface:

- The user does not know that he could use gestures right now.
- The user knows basically about gestures but does not know that this abstract command is possible as well
- The user know that a gesture for his specific command exist but simply can not remember it
- The user ignores gestures generally because the computer will not recognize them anyway
- The user drew the right gesture but nothing happened. No sign of what kind of error occurred.

One important fact is necessary to keep in mind when gestures and their behavior is analyzed: Gestures needs to be learned first. This does not only involve the basic form of the gesture it also involves the commands which can be invoked through gestures. These gestures behave similar to the popular keyboard shortcuts. Both need to be learned and at the specific moment both need to be chosen over the existing menu command. Keyboard shortcuts are only used for accessing the most frequent commands. Instructions which are used rarely will mostly be accessed through the menubar or an icon in a toolbar. If the digital annotation system requires gesture input for common annotation shapes it might work for the frequently accessed commands but John needs a complex set of correction symbols and remembering a gesture to simply input a correction shape demands the double amount of shapes to be remembered. If the users need to look up gestures in the manual to simply insert some standard annotation marks than the requirement of a seamless annotation experience is not accomplished.

Gestures and keyboard shortcuts are an optimized way to access frequently items from the menubar. An interface which is solely based on 'invisible' input methods requires more attention and time to control and can not be considered as easy to learn.

In Johns persona description it is written: *"If something is not really working he is getting quite fast unpatient. That is also true for learning new tools"*^[1]. And in Christoph persona description is the sentence: *"He is open to new technologies and applications but if their learning curve is too steep it is most unlikely that the new stuff is going to take a place in his workflow. Christoph needs tools which he can rely on and if they don't work as promised it is quite sure that they are going to be replaced real soon"*^[2] Will a tool that has the necessity of learning shapes to control its basic interface a chance to survive in Johns or Christophs workflow?

Besides the awareness of gestures there is another problem with this input form. The keyboard shortcut for example can only have one state at the time—pressed or not pressed. In comparison, a gesture needs to be interpreted by the computer to understand what command the user 'asks' for. This process is due to its unlimited varieties of shapes vulnerable to be misunderstood. John might have drawn the right shape but the annotation system did not recognize the shape as the right one. The consequence could be a wrong command gets executed (misinterpretation) or the computer is stuck in an act of idleness (misunderstood). The feedback of both situations is misleading. The user can not understand or analyze what went wrong. The Gulf of Evaluation is not bridged in the event of a misunderstood or misleading gesture. Feedback of the actual state report in which the application resides in, is very important since it allows the user to plan his/her further actions and follow his/her work/goals towards the application.

In the chapter about the spatial orientation a requirement resulted 'Things stay where you put them'. This is not connected to the absolute position of an annotation. It is meant towards the context the annotations has been created in. If John highlights a passage with a question mark and one of his customers opens the document in different document window size the question mark should still questioning the specific paragraph and should not reside in the absolute position John created

it in. This is a technical problem with gestures. Gestures can be executed everywhere on the foil and thus its resulting shape gets inserted at this position. But how is the gesture connected to its context?

The just observed problems are all based around the 'Pencil' tool. The tool design suffers from a problem known in the software design market as 'Feature Creep'. The following sentences has been quoted in the requirement summary already: *"Elegant products don't do everything; but what they do, they do extremely well. What's more, they make it look easy"*^[3]. This limitation is creating simplicity. Simplicity in the software design guarantees that the user knows about the abilities of the software and what is even more important he/she knows how to use them. This is where the 'Pencil' tool fails. The pencil tool tries to provide everything but nothing really well. The pencil tool is simply not elegant. From an interaction design point of view the pencil tool hands out the problems with the input of free form ink shapes to the user. It should have been the job of the interaction designer. The functions of the pencil offer everything but do not take care on the 'How'. That is a common interaction weakness. 'Feature Creep' does not only involve the function design of the whole software it also involves the design of every tool. This thesis claimed already quite often that a software should not only show what it can do, it should also clearly communicate where its limitations are. This is also valid for every tool. The 'Highlighter' tool has straight restrictions which makes the use of it clear. The 'Pencil' is trying to be an all-in-one wonder without any strict usage pattern. Saying that the pencil is for the shapes and marks which the 'Highlighter' or 'Note-Taking' tool do not provide is not the right fundament for a tool design.

Now that the 'Pencil' tool is based on obvious design mistakes there is one more issue with the current tool selection. The research showed that 89% of the students annotations were anchor-only. By generalizing this number the following conclusion would be valid. 89% of the annotation time would be spent on the 'Highlighter' and 11% of the time on the 'Pencil' and 'Note-taking' tool. Does the highlighter provide a good user experience? The 'Highlighter' provides a solid frame in which to use it. The only issue is that it is highlighting only. There is no anchor-only underlining possible. Anchor-only annota-

■ [1], [2]

compare Chapter 'Goal Directed Design' page 88

■ [3]

Peter Blickford, Interface Design: The Art of Developing Easy-to-Use Software, p. 23

tions can have a variety of appearances and forcing the user to mark up the document with a highlighter might be too limited. The research only mentions 89% anchor-only but this does not make clear how many of these anchors were from the appearance 'Highlighter'. The question that arises is why used the interaction designer a 'Highlighter' and not an 'Underliner'? Self-referential design?

These personas really helped to discover the workflow issues inside the current annotation solution. The system itself could work, but its interaction is far away from being as fluid as demanded. The layer is still the key in the annotation system, the sketchy look still reasonable only the user input needs to be revised to optimize the handling to the attention focused active reading which is common on paper.

12

Revising Neon 1.0 to improve the user input issues discovered in the persona analysis. Introducing a new input model for the digital annotation system.

NEON 2.0

THE NEW INPUT MODEL

12 The personas analysis showed that the current annotation input model does not provide the user experience that people are used to from annotations on paper. In 1974 Ben Shneiderman, professor of Computer Science at University of Maryland build up the term **direct manipulation**. Three elements characterize this model:

- Visual representation of the manipulated objects
- Physical actions instead of text entry
- Immediately visible impact of the operation

The key element in direct manipulation is rich visual feedback. “*When I use a direct manipulation system—wether for text editing, drawing pictures, or creating and playing games—I do think of myself not as using a computer but as doing the particular task*”^[1]. That statement follows the argumentation in the previous chapter. Direct manipulation is a very popular and also called First-Person interaction. The user has the feeling of being in control. The computer itself gets invisible and the main task into the foreground. That is why the Apple Human-Interface-Guidelines states out: “*Direct manipulation is an example of an implied action that allows users to feel that they are controlling the objects represented by the computer*”^[2]. But on the other hand Donald A. Norman writes “*Direct manipulation, first person systems have their drawbacks. Although they are often easy to use, fun, and entertaining, it is often difficult to do a really good job with them. They require the user to do the task directly, and the user may not be very good at it*”^[3]. And this is actually the problem with direct manipulation—it is about the task and its control of the items. The mouse is the most common input device. It is great for control-

ling and modifying items which have a reasonable target space. But handwriting or fine motor input is with the gross control of the whole hand covering the mouse very ineffective.

The interaction design draws a distinction between the direct manipulation and the **indirect manipulation**, or **command mode**, or **third person model**. This model has its origin in the beginning of the computer industry while the user needed a special command language to provide him/her with so called commands which the computer then executes to fulfill the users demands. Donald A. Norman uses the comparison of sitting in a car and a chauffeur drives the user to the desired destination while in the the direct manipulation model the user itself is being the driver.

The third person or command mode model has a big advantage: The computer is doing the job which he has been told in a very accurate and repetitive manner. The user gives a command and the computer executes it. The key question for using this model for a very attention effective interaction is how does the user tells the computer what to annotate and how it should look. How does the user gives the computer commands?

The attention gets consumed through the control of the pointing device^[4] in connection with it the rich visual feedback. As stated out in the beginning of this chapter, there are three main properties required for the direct manipulation model.

- *Visual representation of the manipulated objects*

The annotation and the selection needs to be visually represented. Without this representation the annotation system would not make much sense.

■ [1], [3]

Donald A. Norman, The design of everyday things, p.185

■ [2]

Apple Computers Inc., Apple Developer Connection: Apple Human Interface Guidelines, p.34

■ [4]

compare Subchapter 'The Toolbar' section 'color choices', p.79

- *Physical actions instead of text entry*
As noted in the previous chapter where gestures and shortcuts got discussed, shortcuts can be very effective but only for the frequently accessed items. A system solely based on shortcuts or gestures might be very hard to control since all commands needs be remembered because of their invisibility. And having a prompt somewhere on the screen asking the user for some command input is out of question as well.
- *Immediately visible impact of the operation*
This item is responsible for the successful use of a direct manipulation system but it is also responsible for taking up quite an amount of attention.

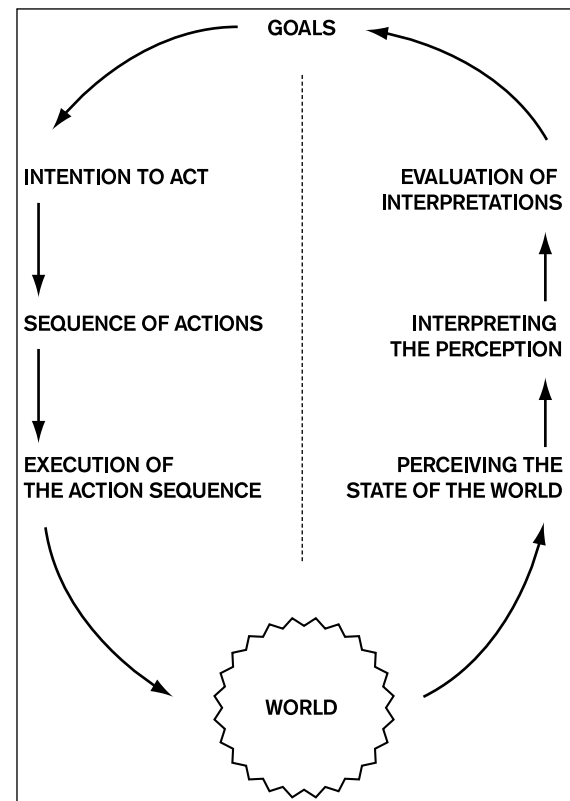
The digital annotation system input model should be a compromise and thus being in between the two extreme input models of direct and indirect manipulation. The user should see a visual representation of his input, he/she should perform physical action instead of entering complex commands or remembering shortcuts/gestures. But he/she should not be forced to draw the items on his/her own. He/she should tell the computer what he/she wants which then will be correctly drawn, placed and connected to the specific shape in the right context. The question that arises now is how to provide the user with a list of shapes? A huge library with all kind of different shapes is not going to add simplicity or less attention consumption. Remember: 'Making choices requires attention.' The key to the interaction experience will be the underlying categorization scheme used to offer the user with a limited amount of shapes.

"Annotations on paper are highly individual in form"^[1]. This makes the choice for groups of shapes even harder. But reading the research over and over again finally revealed that annotations are not so complex at all. The complexity is created through the different visual appearance people uses. Person A likes to work with a pencil and prefers to underline words while Person B makes dotted lines and Person C simply uses a highlighter. Every person has a unique style in annotation. This style is expressed through the visual representation and this is exactly the mistake the first digital annotation system made. It is not the visual appearance which the tools should be able to copy it is the intention of the user. What does Person A, Person B or Person C have in

■ [1]

Catherine C. Marshall, Annotation: from paper books to the digital library, p.139

common? They all want to highlight a specific word or passage. The look of the highlight is secondary. The important part is the users intention and that is in this case 'mark up the word/passage/paragraph'. The efficiency of a digital annotation system is partly depending on the amount of attention the annotation process consumes. Copying visual appearance to achieve the same intention consumes more attention than annotating on paper. The reason behind that lies in the way human beings do things. Considering Donald A. Norman, there are 7 stages of actions:



■ FIGURE 12.1.1 Seven stages of action by Donald A. Norman

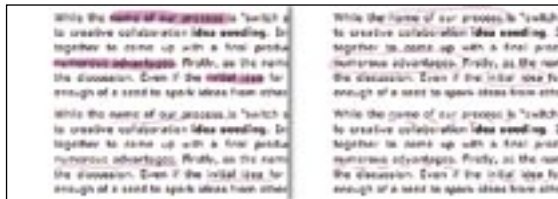
In the beginning of a users action is the goal. The next step requires to move or manipulate something/someone to actually fulfill the given goal. And the last stage is the evaluation of the manipulation to see if the actual goal is achieved. This is known as the basic circle of action. Looking into the several steps in more detail reveals that the execution of a goal requires the separation of the goal into intentions which then gets separated into commands to follow the intention and which then gets executed in a physical manner. The reason why copying the visual appearance with a digital pencil tool takes up

more attention than in the real world is that the execution cycle is the same as in the real world only that the physical action—drawing with the mouse—is far more uncomfortable than actually drawing with a real pen. That is why the interaction design should already start with modifying the execution cycle when the intention is formed. The execution cycle needs to be digitally adapted to work efficiently. The goal remains the same and the intentions to reach this goal as well but the commands and thus the physical action to reach the given intention is necessary to adjust to digital properties. Copying the visual appearance and let the user alone with the last step in the execution cycle will not result in an effective digital user experience.

Now that it is clear that the users intention is the key to categorize the 'shape library' the question that follows is how to create those categories. The research from Catherine C. Marshall reveals already a basic categorization scheme ('Anchor Only', 'Content Only' and 'Compound (Anchor+Content)'). By modifying this scheme into Anchors, Emphasizers, Connectors, and Text the shape decisions should be more easy.

Anchors:

This is the most basic type of annotation. A simple highlight, underline or a circled word is an anchor. The user selects a word/sentence/paragraph and then assigns this anchor an appearance.



■ FIGURE 12.1.2 Examples of Anchor-only annotations

In the above pictures there are some anchor examples. The selection is the same but the look differs. That is the fundament the new input interaction is based on. Every annotation excepts text only inputs starts with an anchor. The anchor is the most basic and as the research revealed the most common type of annotation (89%). The digital annotation systems main focus is the anchor only annotation. If a user starts annotating a text the first selection—look he/she chooses is going to be the default look for the whole document. The next selection is automatically assigned with it. So if the user prefers to underline text pieces the first time he selects a word the user

uses the underline appearance for his selection. All the following selection will automatically be underlined. This behavior is reasoned in two facts the research revealed. First 89% of all annotations are anchor only so in 89% of the selection cases a simple anchor appearance (like an underline) is going to be the right choice. Second the research showed that it is very rare that users change tools during an annotation session. The research concluded that the user does not want to be interrupted in the reading flow. The digital annotation system provides the same behavior. If the user would just like to underline or highlight some pieces of text he is never going to be asked for an appearance again. He/she simply selects the words/sentences/paragraphs and the default appearance gets assigned too.

The digital annotation system allows the combination of anchor appearances. So one selection could be highlighted, underlined and circled for example.

Emphasizers:

Emphasizers are an addition towards an anchor.



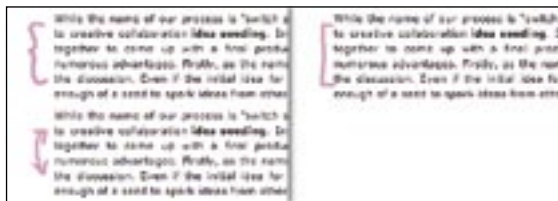
■ FIGURE 12.1.3 Examples of Emphasizers

If the user would like to accentuate a given anchor he/she can add an emphasize to the given anchor appearance. This could be a straight vertical line along the selected paragraph, or this could be question- or exclamationmark or just some asterisks. Again the wording is based on the users intention.

In the real world the user needs to decide about the look of the emphasize out of a pool of infinity numbers of shapes. In the digital annotation system the shapes are already categorized and the emphasize category provides maximum three different shapes. Again this decision is reasoned in the workflow. The user should not worry about the annotation appearance. He/she focusses on the text and the digital annotation system helps him/her getting through it as non distracting as possible.

Connectors:

Connectors are an advanced annotation technique. Connectors can, as the wording already suggest, connect annotations. This connection is not made from a data structure point of view this connection is made from a visual point of view. If for example the user would like to point an arrow from one paragraph to another he/she selects the first paragraph clicks in the connector category on the arrow button. And then selects the paragraph the arrow should point to. The digital annotation system decides about the arrow size, position and direction. The user only needs to worry about the 'What' but not on the 'How'.



■ FIGURE 12.1.4 Examples of Connectors

Text:

Besides all the graphical annotations there is also the need to provide users with the possibility of writing down notes. In the first Neon approach the user clicked on the 'Note-taking' tool in the toolbar and then inserts somewhere on the foil a prompt to start entering his notes. In the new interaction model there is no extra tool for text insertion. If the user would like to add a note he/she simply starts typing. The drawer as explained in a chapter earlier is still present. It will be the created space where the users notes are displayed. There are actually three different situations where the text can be added:

- *The user has selected nothing:*

The text will be added at the top of the drawer with no anchoring or referencing. These are simply some notes on some text. The user can find these unanchored notes on top of each page. This could be useful for To-do lists or general comments towards the text.

- *The users last action was a single anchor selection:*

The text will be automatically anchored to the last selection. If the window gets resized or the user scrolls through the page the text will always stay in line with the anchor it is connected to. Could be used to add something to a word (explanation,...) or a comment to a whole paragraph.

- *The users last action was a connector creation:*

The text is going to be connected to the anchor as well but with a visual connector (arrow, brackets, line, ...) in between. Again, if the user resizes the window or scrolls through the text the text stays in line with its anchor and connector.

Figure 12.1.5 on the following page visualizes Neons new annotation categories and their relation to each other.

This complete categorization scheme is based on text selection. But what happens in situations where the user can not select text? In this case the user will be prompted with the free form ink pen. If Christoph for example would like to annotate a layout and he would like to make clear that the text columns are too wide or that the navigation is placed falsely he can simply draw onto the layout without using the categorization scheme. The categorization scheme only works with text selections and this is where it is very useful. Text can reflow and that is why a strict anchoring of annotations is necessary. Images can not reflow they can only be scaled and that is why the anchoring of free form ink annotations is much simpler than anchoring annotations on dynamic text.

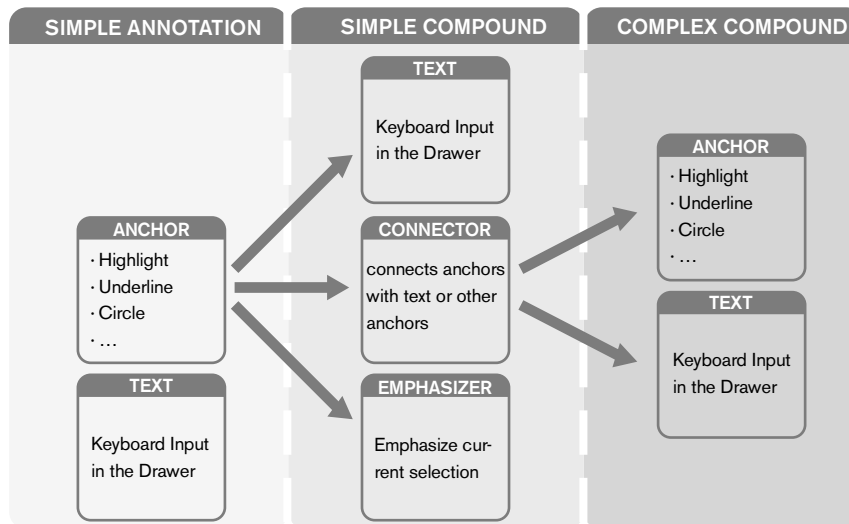
The first Neon approach provided the user with a four tool toolbar. The interaction model was solely based on the direct manipulation model and ignored the users input and annotation behavior. This new approach is a one tool interaction system which supports the user through its annotation process and provides him with the right tools in the right situations. During reading a text the user gets annotations solutions offered. In case the user would like to annotate a diagram or image the user gets automatically provided with the free form ink pen to add any kind of shapes onto the new foil. This system is based on the personas goals and interaction properties. Christoph with his creative background and unique workflow will have all the freedom to annotate given layouts while John and Thomas will get computer aided annotation possibilities to focus on their text and read actively through it.

Level of importance

Annotations are due to their creation procedure highly individual in its form. As mentioned before this is originated in the different categorization expressions. For example Person A likes to underline and Person B likes to dashed underline. Another factor why annotations differs from person to person is the different expression of the

level of importance for each annotation. If the user would like to make clear that this sentence is really important he/she might underline it more intensely than the other words. In the first neon approach the user needed to solve this with the digital version of the free form ink pen. In the new version each appearance (anchor, emphasize, connector, text) has five different appearance levels. If the user hits the button once the standard appearance is applied. If the user hits the button again the underline for example is made a little bit more thicker and the color saturation is raised. There are five steps of this importance levels. If the user hits the last step and clicks again he/she will be at first step again. There is no back and for there is just a single clicking through it. Once more this is designed with the reading flow in mind. The user should not be distracted with targeting extra buttons to increase or decrease the level of importance for each tool or the

user should not remember any shortcuts to change the level of importance. He/she simply uses the same button and just keeps on clicking until the right level of importance is reached. The decision about the right level of importance is not based on numerals. It is the relation to other annotations. The user does not decide to switch directly to level number four he/she just wants this underline to be thicker than the previous one. This relation decision allows this simple type of interaction.



► FIGURE 12.1.5
Annotation-structure

PALETTES & TOOLBARS

12.2 The first version of Neon was characterized by a toolbar which provided the main controls and a menu-bar item for overall control of the annotation system. The new input structure demands a new layout of these controls. Neon went from a four tool interaction system to a one tool system with a categorization scheme. Since a toolbar attached to the titlebar of a document window is flexible in its width, it is not the right space to put the categorization buttons into it. The flexible space does not guarantee that all annotation buttons will have the appropriate space. As stated out earlier contextual menus or menubars hide the tools on the first sight and these are thus invisible. This is not going along with the fast and seamless access of these items the digital annotation system demands and that is why a floating toolpalette (Figure 12.2.1) is the only right choice for the annotation shapes.



FIGURE 12.2.1
Annotation toolbox



FIGURE 12.2.3
Author Info

and in the requirement summary as well, collaborative authoring or sharing of annotation is an issue in a digital annotation system. This sharing feature goes along with the impact annotations could have on unbiased readers. The research in chapter seven uncovered that the anonymity of the annotation author could change the readers view of the virtual audience. Providing an author info should reduce the interference the anonymity could cause. Thus the toolbars function transforms from providing tools to annotate the document into providing information and controls about the present annotation layer.



FIGURE 12.2.2 New toolbar on document window

The first item in the toolbar is the authors name. It is displayed in a pulldown since one document could have several foils attached to and this pulldowns provides the option to select the appropriate foil. The pulldown also serves for the creation of a new foil or to reply to one or to delete foils.

Author info

The new version of Neon provides author info together with each foil (Figure 12.2.3). By pressing the little icon next to the authors name the annotation floating palette will change its appearance from the toolbox to author info. That is why each public foil provides information about the annotators attitude. It includes his favorite RSS

Is the toolbar vanished now? No a toolbar is still present but gets a new function. As stated out in the research

Options Options button

Feeds, music informations or holiday locations.

In the first version the overall controls for the digital annotation system were put into a systemwide menubar item. That included the general activation for the annotation layer as well as the printing feature or the 'Clear all' command for example. In the new version this is integrated into the floating palette paradigm as well.



FIGURE 12.2.4
Optionspalette

There were two reasons for vanishing the menubar item:

- *Awareness of the annotation possibility*
This topic is covered some paragraphs later.

- *Linking of functions the actual foil*

The problem with the menubar item was that it communicated on the one hand to be an overall control item for the digital annotation system but then again it contained functions which were related to the actual visible foil (print, clear all, ...) In the new approach these functions are grouped inside the floating palette. This communicates much better that the displayed commands which relates to the actual visible foil only.

Minimize Floating Palette

This icon provides functionality for hiding or showing the floating palette.

Open System Preferences

This icon is opening the overall system preference pane for the digital annotation system. This has not been covered in the first version. The user should be able to customize some of the annotation systems behavior.

As the research uncovered annotations are highly individual in its form. It is very hard to determine which shapes would serve most users well and then again by looking at

the research shapes differs between the text topic (scientific text, fiction reading,...). As Catherine C. Marshall wrote: "*The forms of annotation are also clearly shaped by disciplinary expectations and text book genre*"^[1] it is difficult to limit the kind of possible shapes. That is why the annotation shape floating palette is being able to get customized. John uses different annotation symbols than Thomas for example. Thomas does not make his living by correcting or translating technical documents he simply annotates texts for his study or mark up passages in his personal blog. The shape palette has a default look which is going to serve most users well but for special purposes users can fill the floating palette with shapes from a big library in the system preferences.

Other items which are open for a customization discussion could be color choices or the general foil color or default export options or print layout settings and so on. These settings needs to be determined through extensive user testing and through political decision by the software vendor.

Awareness of the annotation possibility

As mentioned some paragraphs earlier the overall menubar item in Neon 1.0 did not communicate its application independent functionality and presence. The menubar offers overall system controls as well as application dependent menus to provide the user with specific features. But in the demonstration of this annotation system solution none of the audience members recognized the annotation feature. The menubar was only recognized when a problem occurred or the user looked for a feature. The initiative was on the user side. Besides its not very prominent visibility it was not clear if this was an application feature or something provided by the operating system. That is why the menubar item needed to be moved at a location where it clearly communicates its basic behavior of being a new property for **every** document window. The new place is in the windows titlebar where all single window properties are gathered.



FIGURE 12.2.5 Neon button in titlebar

[1]

Catherine C. Marshall, Annotation: from paper books to the digital library, p. 135

13

How do John, Christoph and Thomas achieve their goals with the digital annotation system? Scenarios describe personas using software.

SCENARIOS

13 Personas are a model for Goal-Directed design. The reason why somebody interacts with something is the presence of a goal. To achieve this goal the person has to complete several tasks. It is important not to confuse tasks with goals since it is very easy to mix them up. Goals represent an end condition while tasks are “*intermediate processes*”^[1] to achieve such goals. The essence of good interaction design is to give the user possibilities to successfully realize his/her goals not just their tasks.

One property which differs goals from tasks is that tasks changes as technology changes while goals mostly remain stable. For example if somebody wants to travel from Italy to Sweden his/her goals might be speed, comfort and safety. In the 19th century he/she might have taken a horse plus a wagon for comfort and a gun for safety. Today he/she might consider an airplane for speed and comfort and let the gun at home for safety reasons. Tasks are needed to achieve goals but software needs to be designed for fulfilling goals not tasks.

Since the goals for the personas in this thesis are known the next thing that is to determine which steps are necessary to fulfill these goals. The appropriate tool for incorporating tasks is called scenarios. “*A scenario is a concise description of a persona using a software-based product to achieve a goal*”^[2]. Building scenarios is often compared to acting. The interaction designer needs to inhabit the character he/she has to play. He/she needs to feel, think and act the same way as the person he/she is about to play. The designer needs to forget his/her own interests, his/her education and personal experience and really imagine to be someone else. While building scenarios out of the given personas the designer needs to behave similar. The scenarios needs to reflect as exact as possible the way John, Christoph and Thomas are going to achieve their goals. It is not important to visualize each little task but it is important to build the scenario from the beginning to the end to fully show the way these personas achieve their goals. During

this process it is not the main focus to split a goal into a lot of small task pieces. The most important thing is that the goals are achieved and that it has been tried to eliminate as many tasks as possible to fulfill these goals.

■ [1]

Alan Cooper, *The inmates are running the asylum*, p. 150

■ [2]

Alan Cooper, *The inmates are running the asylum*, p. 179

JOHN

Translator

13.2 Johns main goal in using the digital annotation system is to be more efficient in the electronic document handling. His scenario is about reviewing text and send it to the customer.

Text correction/translation in general involves a lot of coordination with the customers. In the days where paper was the only text based communication medium the professional participants developed a coded correction language to avoid misunderstandings throughout the correction/translation process. These symbols are a special form of annotation. In the beginning of this interaction design process it has never been imagined to integrate these annotation signs into the digital annotation system since it would probably only be helpful for a fairly small amount of users and would complicate the whole interface through its complexity (80/20 rule). But the area of correction/translation is something where annotations have a major impact and that is why they have always been in mind. Neon 1.0 had the free form ink pen which allowed the user to draw all kinds of shapes. But these correction shapes are too complex to sketch easily with the common input device – the mouse. Thus the first solution offered the user with gestures. These needed to be learned to finally create these special correction annotations. But the personas showed that this solution is not going to serve the users needs. *“John understands his computer mainly as a tool. If something is not really working he is getting quite fast unpatient.[...]You could say that John is sort of a manual guy, abstract workflows (own scripts or actions) aren’t his preference.[...]His passion is not in the computer itself, his passion lies in the outcome”*^[1]. Telling John to learn abstract gestures would be a failure in the software design.

That is why Neon 2.0 focusses on the intentions of the user. Johns special annotation symbols can easily be translated into the system of anchors, emphasizeers and connectors. The floating palette can be customized to meet John demands. The basic interaction is similar to the Microsoft Office Suite where for example John needs to select a word/paragraph as well to assign a character format like bold or italic. This similarity should make the interaction with the digital annotation system less disturbing and more fluid since John knows the Microsoft Office Suite quite well.

The customization of Neon is a system wide modification. One of Neons main characteristic is the application independent functionality. To be consist with this functionality it is important that all modifications are applied system wide. That is why the customization takes place in an application independent 'system preferences pane'.

■ [1]

compare chapter 'Personas', p.88



FIGURE 13.2.1
Standard annotation palette vs.
customized annotation palette

There is one behavior which is different compared to standard annotations. Correction signs appearance needs to be 'repeated' in the margin space. So if John applies a connector between two words which uses the appearance for wrong word order, this appearance needs to be repeated in the margin. In the case of the digital annotation system that would be the drawer.

The following screenshot shows the feedback John is making towards some changes the customer did. The application independent design makes Neons appearance consistent throughout the whole digital work environment. The customer will be able to see the annotations inside Microsoft Word (independently which version he/she uses) or any other appropriate Word document viewer. There will be no difference since the annotation has nothing to do with the Word file itself. It is important to understand annotations not as exclusive applications meta data but more as content meta data.

John is reviewing a catalogue from his customers with the special annotation symbols. As described above the correction signs are repeated in the drawer.

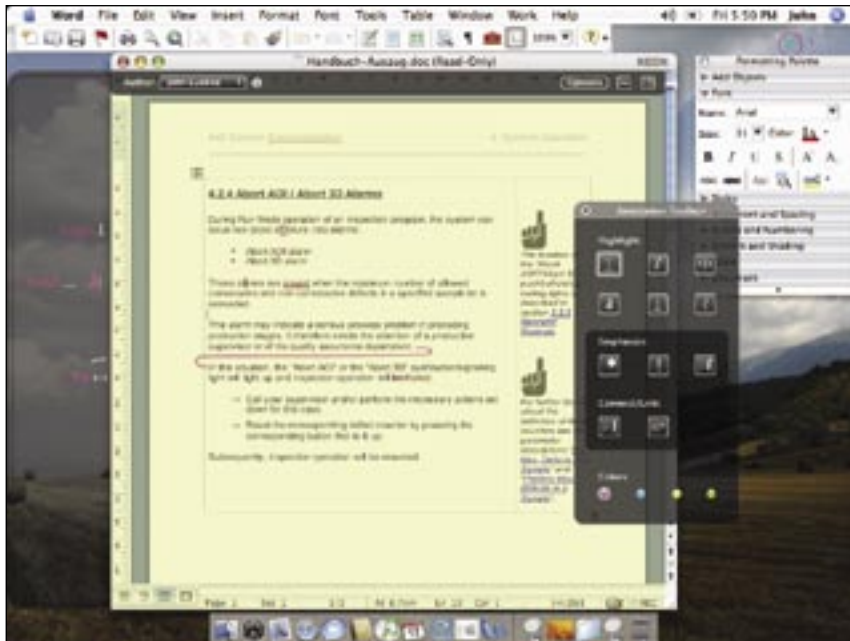
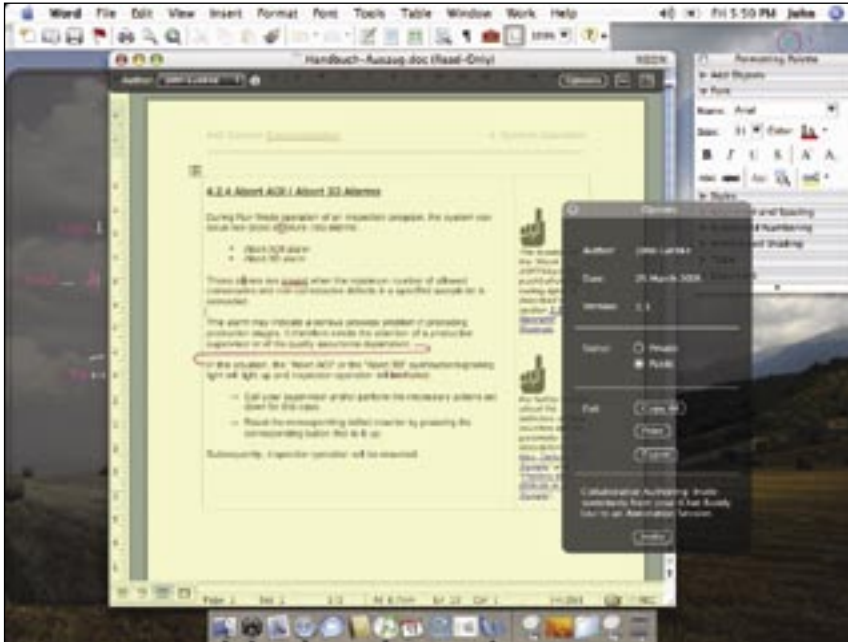


FIGURE 13.2.2 Neon in Microsoft Word

The next screenshot shows how John needs to make an effort to declare a foil as public (radiobutton in the option palette). This enables the sharing of this foil.



► FIGURE 13.2.3 Setting a Neon foil as public

The following screen shows the notification of annotations inside the Apple Mac OS X Finder (icon changes). This informs John about the presence of annotations without the need of opening the file in some application



► FIGURE 13.2.4 Icon notification

The final screenshot shows how an annotated file is going to look inside Johns email application. Since John declared the foil already as public and thus put an effort into the publishing of content the annotations are already attached to the file John is going to send to his customer.



FIGURE 13.2.5 Annotated document as an email attachment

These screenshots describe the daily tasks John needs to fulfill to get his work done. The digital annotation systems integrates seamlessly into Johns current workflow.

CHRISTOPH

Art Director

13.3 The work of an 'Art Director' is dominated by reviewing and commenting existing work and communicating the results to the clients. A lot of meetings with sketching and several proposals are common. These meetings are characterized by the contribution of every participant and clear communication is a necessity. Looking at the current meeting habits reveals that it is very common to print out the digital work and draw with a standard pencil on them to illustrate the ideas of the meeting subjects. This is a common usecase where the document workflow between digital documents and real world papers is quite inefficient. The findings of these meetings needs to be summarized and sent to every participant. So the work on the real world paper needs to be transferred into the digital world again for easy distribution. This step could be avoided if a digital annotation system is present.

The following screenshot shows a layout in Adobe Photoshop. Christoph and some designers are going to discuss the results.

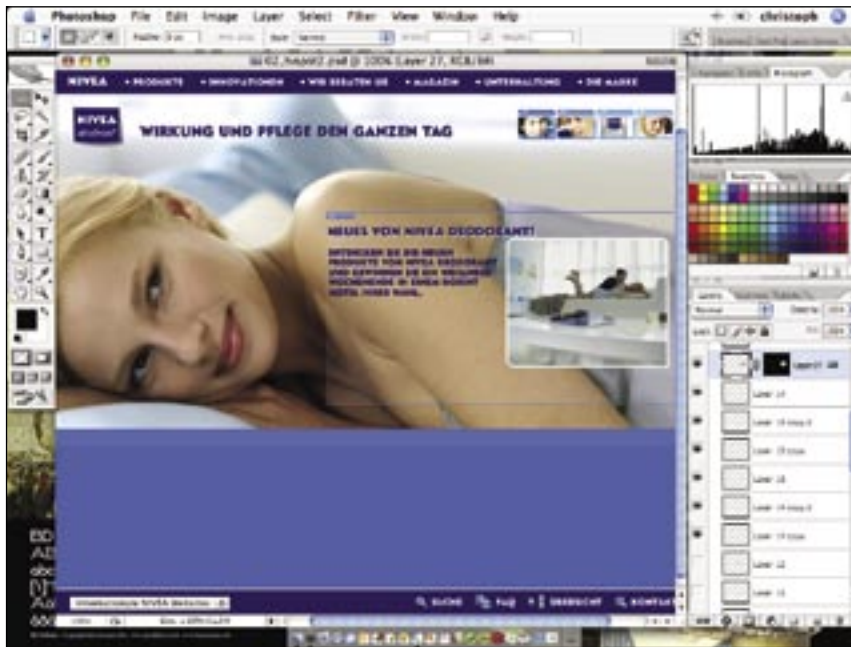
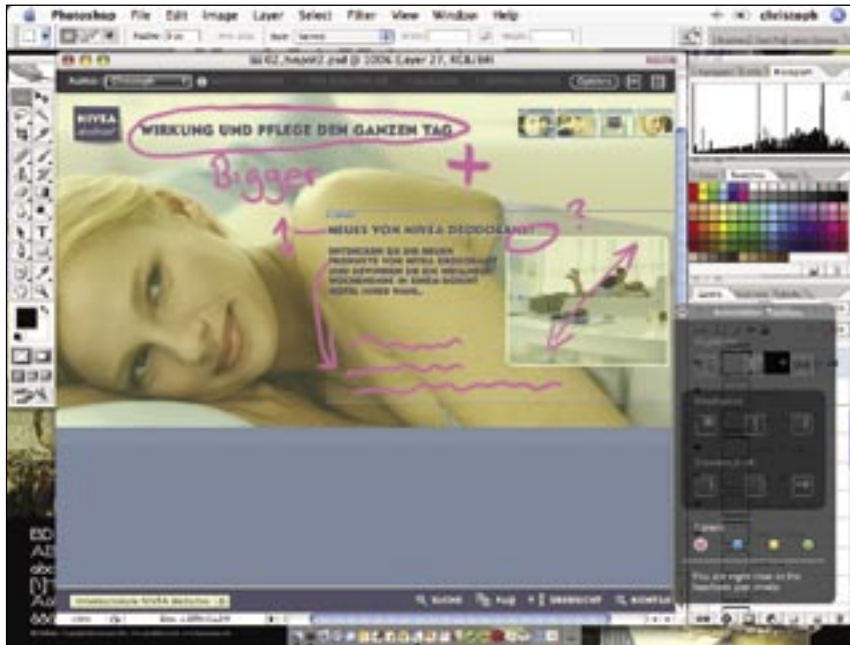


FIGURE 13.3.1 Layout in the Adobe Photoshop environment

Neon's input model is based on text selection. Since the screenshot above clearly shows that there will not be any text selection possible, Neon automatically greys out the annotation tools and switches the annotation into the freeform ink mode. This mode

allows Christoph and his team to work very similar to paper sketching. The mouse input is definitely an issue but most designers are used to draw with the mouse and the need for annotating pictures is definitely not wide spread.

The final screen shows the greyed out annotation toolbox as well as the annotated document.



► FIGURE 13.3.2 Annotated layout with the freeform ink mode

THOMAS

Student

13.4 Thomas' scenario will be split in two separate workflows. The first will illustrate how he annotates a school document together with a colleague and the second scenario shows how he publishes annotations to his own blog.

Annotation session with a colleague

The research showed that annotations are not limited to be only used personally. The IMF (International Monetary Fund) showed that collaborative authoring was a major issue in their daily work and that reviewing of annotations are mostly centered around paper. The participants did not mention the reason for it but other studies revealed that annotating digital documents felt cumbersome. The question that arises is on how annotations could work in a collaborative authoring situation. The challenge is not simply to provide the participants on such an authoring session with a paper equivalent workflow but also communicate the collaborative situation clearly. Apple Computers Inc. introduced in 2002 a chat application called iChat. This application allows people to communicate over the internet or even in the LocalAreaNetwork with a technology called ZeroConf (Bonjour). The communication model is not restricted to text only chats it also allows voice communication similar to a phone call. The iChat application framework is present in many other applications. If somebody receives an email from somebody on his/her iChat buddy list the email application displays the current online status of the email author. This system wide presence and easy configuration makes it a promising solution for collaborative authoring/reviewing situations.

The option panel of the digital annotation system provides the user with a button to invite a person from his iChat buddy list to an annotation session. Thomas opens up an article for his study, invites a colleague to annotate the given text, the text get than transferred automatically to the invited buddy and both can annotate the source text. This can be combined with a text chat as well as a voice chat. All annotations are automatically synced between both annotations partners and the text together with the annotation foil is save locally on both computers.

The first screen shows the document that Thomas would like to annotate with his friend.

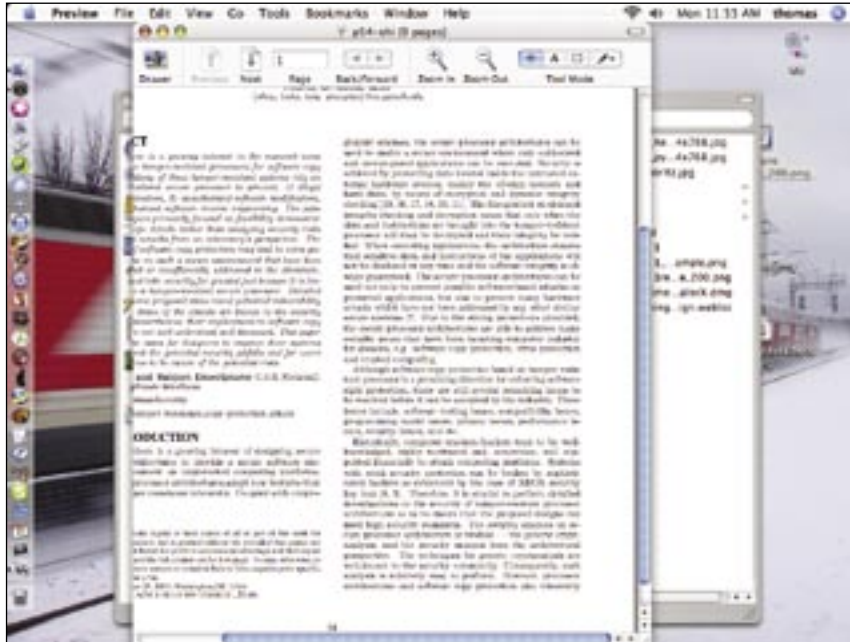


FIGURE 13.4.1 PDF document that Thomas would like to annotate

The next screen shows Thomas' option palette where he can invite his colleague to take part on the annotation session.

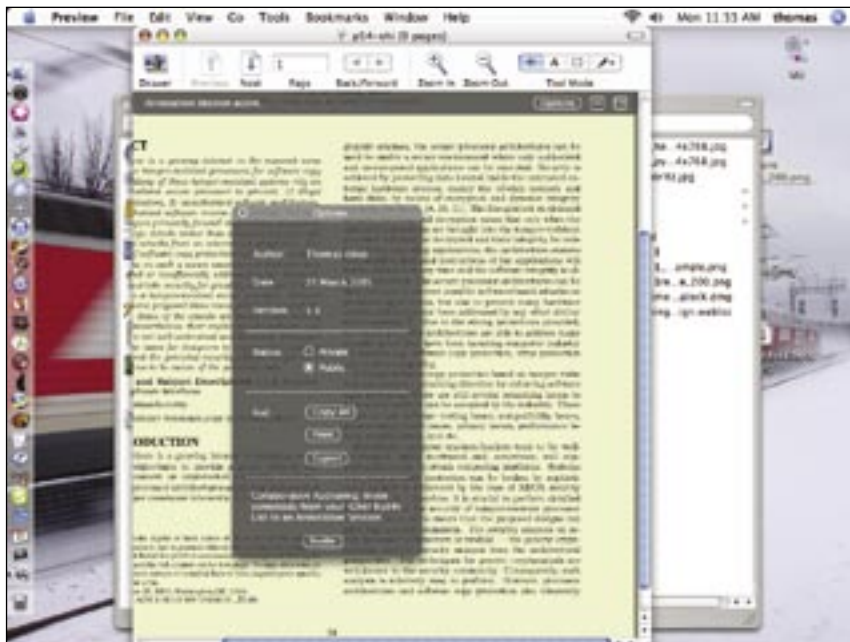
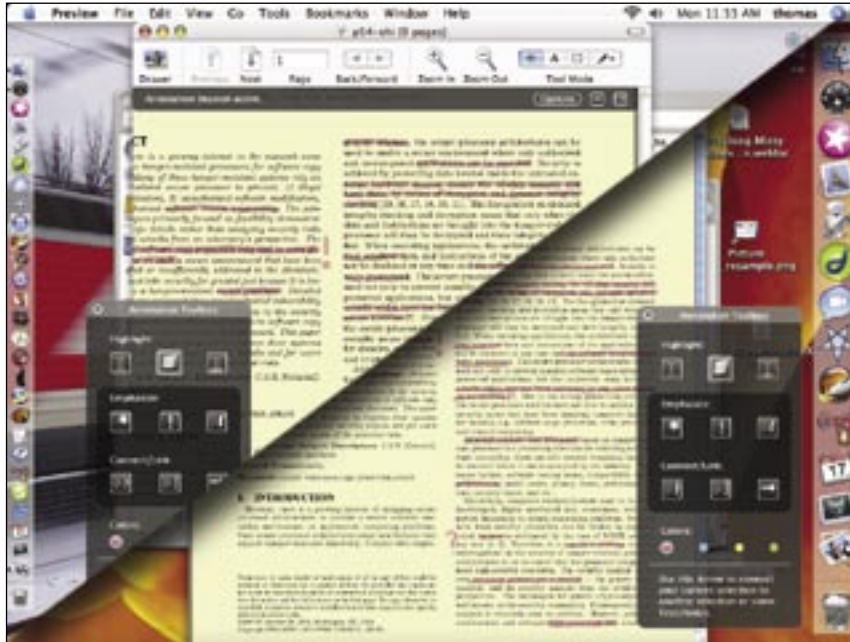


FIGURE 13.4.2 Option palette which allows Thomas to invite his colleague

The final screenshot displays the two screens from Thomas and his colleague annotating the same document using the iChat framework.



■ FIGURE 13.4.3 Split-screen of the annotation session

Publishing annotation together with an article on Thomas' blog

As mentioned in Thomas personas description his communication is characterized to be mainly digital. Thomas uses a weblog to publish articles about his life and his experiences on the web. Some of these articles contain annotations to provide his visitors with context related extra information. This scenario describes the workflow on how Thomas publishes his annotations on a website.

The first screen shows the original blog article together with the annotation Thomas did. Right now these annotation would only be visible on Thomas' local machine.



■ FIGURE 13.4.4 Annotated blog article on Thomas computer

To make the file accessible to all his visitors Thomas needs to upload the annotation foil onto his webspace. This allows everybody to view his annotations. To access the single foil of his article he clicks on the 'Export' button on the option palette. This will save the foil only. The single file gets then uploaded on his webspace. The question that now arises is how will users access the source article together with the just uploaded foil?

The HTML file definition allows the linkage of different files. This is for example commonly used when CascadingStyleSheets (CSS) are associated with an HTML file. A CascadingStyleSheet *"is designed to allow the separation of presentation and structure"*^[1]. This is similar to the architecture of this digital annotation system. Basically the annotation layer is just a different presentation but not a new structure. To associate the uploaded foil with the HTML page which contains the article Thomas published in his weblog he simply links the annotation foil to the HTML page in the same way CascadingStyleSheets are linked to the file as well.

■ [1]

Wikipedia, chapter 'CascadingStyleSheets', subchapter 'Use Of CSS'

The following screenshot shows the source code of Thomas' blog article. The annotation foil is linked in the HTML source.



FIGURE 13.4.5 Addition in the HTML source to attach the annotation foil

If a visitor of Thomas' weblog opens up the annotated article the digital annotation system will notify the visitor about the presence of annotations on this webpage by changing the color of the Neon button in the document window toolbar. The visitor can not modify Thomas article that is why no annotation toolbox is displayed.



FIGURE 13.4.6 Thomas' annotated article on a colleagues machine

14

Since the other annotation requirement list, the solution from this thesis should be compared with the postulated pass this test as well.

REQUIREMENT LIST CHECK FOR NEON 2.0

NEON 2.0

'A digital annotation system for Mac OS X'

■ VENDOR: Thesis at Fachhochschule Lübeck, Germany	■ INVENTORS: Meikel Steiding	■ DATE OF OPERATION: 2005	■ BUSINESS MODELL: Concept/Study
---	--	-------------------------------------	--

14 In the previous chapter the research has been compared to present annotation systems. Now that a new digital annotation system has been designed it is interesting to see if Neon fulfills the requirements which has been postulated necessary.

REQUIREMENT 001

■ ACHIEVED

This requirement was the only one which none of the present annotation solutions could fulfill. Neon has been designed with exactly this requirement in mind. The basic fundament—the layer idea—on which the whole system is build on provides an annotation functionality in every document window. The communication of the awareness of annotation has been a major point in the overall interaction design. The layer does not only provide an all time present annotation possibility it even provides a very flexible and expandable metaphor which creates a complete own workspace for the annotator.

REQUIREMENT 002

■ ACHIEVED

The workflow has been a major issue. The first Neon approach has been very typical for a computer application. it was focused towards the direct manipulation paradigm and prompted the user with a tool design that was orientated on the real world behavior. That resulted in a tool which allowed every shape to be drawn but ignored the common input method completely. The appearance was copied but the annotators intention not even considered. That is why a new interaction model has been designed and so far this looks really promising. It has been inspired by the research on annotations and focusses directly on the tasks the annotation is split in.

REQUIREMENT 003

■ ACHIEVED

The first Neon design started with a four-tool interaction concept. This seemed an average approach and was in comparison to other annotation systems already a good solution. The new Neon design focuses on a single-tool model. The input-interaction remains the same and only the appearance differs. For simply highlighting a text and writing some notes in the margin there is no tool changing involved. This is a huge improvement in comparison to all existing annotation solutions which have separate tools for all these tasks.

REQUIREMENT 004

■ ACHIEVED

All the basic annotation types are possible. Annotations are based on simple text selections where an appearance is assigned to. This allows all kinds of different shapes which the user can add into his own floating annotation toolbox. If no text selections are possible (i.e. diagrams, images,...) the user is automatically provided with a free form ink pen which allows all kinds of different shapes as well.

REQUIREMENT 005

■ ACHIEVED

As mentioned in requirement 001 already, annotations take place on a complete separate layer. This layer does not only provide an own data level or an own workspace it also distinguishes contents and annotations. The layer idea does not only communicate the distinction of content and annotation it also puts the content in the main focus and lets annotations be annotations, namely data about data. This is something which for example the Adobe Acrobat reader completely ignores.

REQUIREMENT 006

■ ACHIEVED

Since every annotation layer is a complete own workspace every author is provided with its own foil. There is no chance that someone could destroy the annotations of another author. This is not realized through interaction with the interface it is simply the nature of the used metaphor. This makes an explanation towards the user unnecessary. *"When affordances are taken advantage of, the user knows what to do just by looking: no picture, label, or instruction is required"*^[1].

■ [1]

Donald A. Norman, The design of everyday things, p.9

REQUIREMENT 007

■ ACHIEVED

The annotation layer is only a new property for the document window. The user does not leave the current application or needs to open a specific plugin. The save command in the 'File' menu would still save the source content of the document window. Each new foil attached to the document window gets automatically saved in the background. There is no 'Save annotations...' button. It is exactly the same behavior as on paper, where every action is automatically saved as well.

REQUIREMENT 008

■ ACHIEVED

This has been one of the reasons why the first digital annotation system has been re-designed. The new version focusses on anchor-only annotations since it is the first and most basic type of annotations in the new categorization scheme. Every text annotation starts with an anchor.

REQUIREMENT 009

■ ACHIEVED

The same 'engine' in which the user creates the annotations is used for displaying the annotations. The markings could be stored in a xml-like format. That means that the annotations itself are not stored inside the file but rather the description which words/sentences/paragraphs needs to be annotated. When the file gets loaded the annotation system interprets the description inside the file and places the annotation information onto the foil. Since 'creation engine' and 'displaying engine' are the same, the look and behavior is equal.

REQUIREMENT 010

■ ACHIEVED

There are mainly two places where the presence of annotations is communicated. First in the closed file state, where the user is prompted with a new icon and secondly in the file opened state where the Neon icon in the window titlebar changes its color. This will make it very clear if there is more to a file than its main content.

REQUIREMENT 011

■ ACHIEVED

Neon has a sharing feature build in. The user needs to declare a foil as public to make it sharable. If no foils are declared public and the user puts the document on a server or sends it via email the associated foils will stay local. If foils are declared public the interface will inform the user about which foils are going to be transmitted.

REQUIREMENT 012

■ ACHIEVED

The data a foil file contains is text only. Everything is based on real time interpretation when the annotation is viewed so the file size is kept as small as technically possible.

REQUIREMENT 013

■ ACHIEVED

As mentioned already the most basic type of annotation in Neon is the anchor. As the name already explains the annotation is always connected to a text selection. So even if the text reflows through window resizing or text deletion the annotation will contextually stay where the user put them. On images which can not reflow the information is stored proportionally from the top left point of the annotated image. If the image changes its position the annotation will follow since the anchorpoint is not in the top left corner of the source document it is the top left corner of the annotated image.

REQUIREMENT 014

■ ACHIEVED

The floating palette position and the last activated annotation shape will be stored inside the foil but are limited to work locally.

REQUIREMENT 015

■ ACHIEVED

As mentioned already in the sharing requirement. Each foil has a property of its publication state – private or public. The user needs to declare a foil as public before it is possible to choose from in the sending file dialog.

15

Plans for the future?
What are the results?

FINAL WORDS

FUTURE SIGHTS

15 This chapter will present three obvious development steps for future Neon versions. All of them are not new or revolutionary. They are a logical sequel to the current solution and could greatly enhance the whole user experience.

Web plug-in and javascript activation

The success of a digital annotation system is depending on its distribution. As noticed earlier in this thesis collaborative authoring and thus sharing of files is an issue in the annotation process. If the user can not assume that the communication partner is able to view the annotation of the files, he/she needs to export the annotations together with the source document into an appropriate file format. But what happens in Thomas' scenario where he annotates his own blog? There are only two solutions in case the visitors of Thomas' website do not have the annotation system build into their operating system. First they are not going to see the annotation at all or second they need to download extra files (i.e. pdf or jpg) to finally view the annotations. None of that is an appropriate solution. Since the whole annotation system–design is deeply linked into the WIMP paradigm it would greatly enhance the common user experience if people could rely on the possibility of taking notes or marking up words in every document window on every operating system. To make this digital annotation solution more popular and to reach a much broader audience it might be a possibility to give people who do not have this annotation system build into their operating system the chance to download a web-browser plugin to simply view annotations from others on their systems. If Thomas is going to publish an article with annotations attached to, users of this annotation plugin would get a button which would notify them about the presence of the annotations and would then slide out the foil over the webpage to display the attached annotations. The behavior and look of the webbrowser plug-in needs to be very very similar to the annotation system

build into the operating system. It might be impossible to place the annotation button in the windows title bar but the animation and the overall presentation of the annotations needs to remain the same. The plugins purpose is only to be able to view the annotations not to create some. The creation process of annotations is complex and this thesis showed how important it is to remain the reading flow during active reading.

Another feature which goes along with the internet presentation of annotations is the ability to activate the annotation layer via interactive links on a webpage. The author of the webpage would get the possibility to incorporate textlinks or graphic symbols into his webpage which then would invoke the annotation layer. Thus the author can actively announce the presence of annotations on his website. A good example could be news websites. Next to the option of 'Print this article' or 'Email this article to a friend' there could be the option of 'View the authors annotations'. That would help to integrate the annotation system into the whole user experience and communicate annotations as extra data to the present article.

Neon atomized

During the daily computer experience especially with an eye towards websites there might occur the case where the user is not able to modify the whole content of a document window. Dynamic websites are a good example for this. If there is the possibility to upload a picture or any other file to a website to contribute something to a web community the user is not in charge of the complete website appearance. The other parts of the site might be modifiable by other users or simply by the author of the website. The basic Neon approach of annotating the whole webpage and then allowing others to view the annotation would not work in these cases since the user is only responsible for a single item of this website and does not own the site. A way to allow more contribution

through annotations could be that the neon paradigm of sliding out a foil and use this space for annotation is becoming atomized onto objects. Single objects like images for example could get the foil property as well. Thus the foil metaphor is not only valid for the complete document window it also works on single items. If a user submits an image to a webpage which has some annotations attached to, other users could activate the foil of the image and view the annotations which have been attached to this image. Thus users could easily comment or give explanation to objects send to somebody.

This would require redesigning the notification system about the presence of annotations, as well as the simple annotation process of these items. But it could have a great impact of exchanging objects with comments or drawings attached to!

Apply correction

Looking at Johns persona description again reveals that he is using annotations generally in a very different way compared to Thomas or Christoph. They mostly use annotations as a notification or remembering medium. But John receives and creates annotation with a different purpose. His annotations mostly imply a suggestion. Suggestion is defined in the dictionary as *"an idea or plan put forward for consideration."*^[1]. John or his customer then decide if the suggested correction takes place and this would result in a correction towards the source document. An improvement to this interaction could be that the suggested correction displayed through the annotation signs could automatically be applied. Neon would then ask the user and if he/she agrees that the text correction is done through Neon. The interaction could be similar to the standard spell checking dialogs. This would improve the workflow of John and his customers. The solutions implies that every single annotation shape will get new properties for defining the way the correction would take place. A lot of research and even more testing of this would be necessary to make sure that this automatism is reliable. It would greatly improve the efficiency of Johns work and how he deals with documents.

■ [1]

Apple Dictionary, Edition Oxford american dictionary, word:suggestion

CONCLUSION

15.2 As mentioned in the introduction of this thesis the requirements for today's Graphical User Interfaces have changed. The computer is much more present in the daily life and thus needs to fulfill different tasks than in 1984 when Apple Computers introduced the first GUI to the mass market. The possibility to annotate documents inside the digital world is still a feature sold by single software vendors. They do not support the active reading workflow from paper and this might be one of the reasons why annotating in the computer is still unpopular. The research in this thesis proved that annotations are still a major paper feature. Paper is mostly used in collaborative situations as well as for reviewing. Today's computer systems provide good tools for text creation and editing but reviewing existing documents is mostly done on paper. The increasing amount of digital documents demands a solution for it.

The interaction design concept in this thesis has been centered around the user's goals and problems. It is not a new application, it is a logical progression to the current WIMP paradigm. This digital annotation system is based on existing interface items and user habits but redefines the popular window metaphor. The document window will get a new property for displaying foils onto the source content. This new content layer results not only in a new data structure it also communicates the key functionalities a digital annotation system needs to provide. The new input model introduced with Neon 2.0 has been solely designed with the annotation habits from the users in mind. It guarantees that the least amount of attention is required to control the interface and that the user can focus on working actively through the text. The features of the system have been selected to allow most of the users to efficiently fulfill their goals.

This annotation solution could be one of the refinements that the Graphical User Interface of today's computers need, to complete the tasks with which users are prompted every day. The computer's aim is to help the user getting his/her work done more efficiently. Printing out digital documents for reviewing or collaborative authoring is not in line with this postulation. A digital annotation system is not going to vanish paper nor is it a concurrence, its aim is to improve the efficiency of working together with digital documents and paper in the real world.

A

CD CONTENT

FOLDER STRUCTURE

001 

Neon 1.0

This video was created to present the basic Neon idea and its features to different companys. An audio track explains the background and single actions on screen. The video can be controlled via the cursor keys [LEFT] and [RIGHT] to skip between chapters. The [SPACE] bar wil play/pause the video.

002 

Neon 2.0

This video is only a small update to the Neon 1.0 presentation. It has no audio track since it was always presented personally and illustrates the new input model introduced with Neon 2.0. This video uses the [SPACE] bar for play/pause as well.

003 

References

All files included with this thesis are copyrighted by their authors. They only serve the purpose to look up the mentioned quotes in this thesis. Distribution without the authorization of the appropriate authors is strictly forbidden. The texts are sorted in alphabetical order as well as listed by chapters equally to this thesis, too.

004 

Thesis

This folder includes a screen optimized version of this thesis in the Adobe PDF format. Distribution of this PDF without authorization of its author is strictly forbidden.

005 

Ustertest

Please copy the appropriate files onto the local hard-drive. This test will output several png images which contain the drawings from the test. They can be deleted afterwards.

B

For Neon 1.0 a usertest possible or which kind of
has been designed to shapes do create prob-
analyse which shapes are lems.

THE USER TEST

B During the first design of the digital annotation system, where direct manipulation was the main paradigm to create annotations on the new document window layer, the question arises which shapes are hard to draw with the common input device the mouse. How difficult is it to draw straight lines for underlining words? Can the user create circles or simple curves? To answer these questions a usertest was set up. This test should also create basic cognitions which shapes could be used for the gestures input. Since the whole input interaction changed during the design process this test never made it public and no results have been captured.

A website was set up to allow a broad audience to download the user test for Apple Macintosh and Microsoft Windows systems. The test was an application build in Macromedia Director. After the test was completed the user would be asked to send the result to the email address neon@playgroundz.com. The test outputs eight 'PNG' images into the same directory where the application is located. These images contained the drawings the user did through the survey.

The test started with seven questions to categorize the test subject. The first two questions were reasoned in building different user categories based on skills and experience.

Q1: Please choose your daily activity with the computer

- I am mostly working in an office suite (i.e. Microsoft Office, OpenOffice,...)
- I mainly use my computer for surfing the internet. (i.e. ebay.com,...)
- I mostly draw/paint/layout (i.e. Adobe Photoshop or Illustrator, Macromedia Fireworks,...)
- I mainly use editing applications (i.e. Macromedia Flash, Apple FinalCutPro, Adobe InDesign,...)
- I am programming most of my time (i.e. vi, emacs, BBedit, XCode,...)

Q2: What kind of input device do you use regularly?

- Mouse
- Trackball
- Trackpad
- penlike input device (i.e Wacom)

The three upcoming questions were asked to understand how important the input device for the user is and how much he/she is aware of it.

Q3: Did you choose your input device on purpose?

- Yes
- No

Q4: Do you feel comfortable with your current input choice?

- Yes
- No

Q5: Would you like to try a different input method?

- Yes
- No

The last two questions were centered around the skill level of the participant to refine his/her user profile.

Q6: How long have you been using this type of device?

- Weeks
- Months
- Years

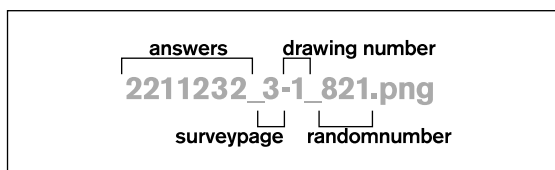
Q5: Are you aware about a variety of Keyboard Shortcuts?

- Yes, my friends call me Mr./Mrs. Shortcut.
- No, I am more into pointing and clicking.

After the user answered these questions he/she needed to draw different kind of shapes which got captured in the background. All these four drawing tests were separated into two procedures. In the beginning the user were asked to draw slow and accurate while on the second drawing panel he/she should try it in a more fast and sketchy way.

The first drawing set dealt with horizontal lines which are similar to underlining or highlighting paragraphs. The second drawing set was centered around vertical lines. In the third drawing panel the user were asked to draw open circles which were inspired by circling words/sentences/paragraphs. And in the final drawing exercise the user should draw waveforms similar to a sine wave. The amplitude differed between the three different waveforms to analyze how user deal with the direction changes. These findings would have been really important for the gesture design.

As mentioned before these drawings were saved as transparent png to allow easy analyzation of a broad audience. The filename of each png also contained the answers from the questionnaire to allow an easy analyzation of the received results:



■ FIGURE B.1 Number code from the test results

Neon: Digital-Ink Annotations

Neon is the first phase of my study in Germany/Denmark. It is a new digital-ink annotation system which provides application independent support for annotations and highlighting. Easy-to-Use and flexibility is gained by redefining some of our existing GUI notations. This test consist of two parts: 7 questions and 4 drawings. It will take less than 5 Minutes to fill this test. Please read the test page carefully. Thank you for your support.

Question.01 Please choose your daily activity with the computer.

- I am mostly working in an Office Suite (e.g. Microsoft Office, OpenOffice, ...)
- I mainly use my computer for surfing the internet (i.e. wikipedia, ...)
- I draw/paint/layout a lot. (e.g. Photoshop, Illustrator, Fireworks, Paint, ...)
- I mainly use editing applications (e.g. Flash, Final Cut Pro, iDesign, ...)
- I am programming most of time (e.g. vi, emacs, bbEdit, KCode, ...)

More info...
These questions are asked to determine which kind of annotation you belong to and how a test is designed your variables. It should give me an insight in your skills and experience and let me know what kind of input device you are using.

Question.02 What kind of input device do you use regularly?


- Mouse
- Touchpad
- Touchpen
- Penlike Input Device (e.g. Wacom)

Go To Next Page


► FIGURE B.2
Example page displaying the questionnaire.

Drawing.04 Please follow these "waves" with your input device ...

a) slow and kind of accurate



b) fast and a bit more steady



Go To Next Page

► FIGURE B.3
Example page displaying the drawing section

C

REFERENCES

TEXT

C.1 Websites, Books, Journals used to support this thesis. All electronic documents were saved in the PDF format and placed on the attached CD. Listed in alphabetic order

- Adobe Inc, What is PDF, 2005, [<http://www.adobe.com/products/acrobat/AdobePDF.html>], 3.9.2005
- Apple Computers Inc., Apple Developer Connection: User Experience/Controls&Cells, 6.11.2003, [<http://developer.apple.com/documentation/Cocoa/Conceptual/TabView/TabView.html>], 3.9.2005
- Apple Computers Inc., Apple Developer Connection: Apple Human Interface Guidelines, 9.8.2005, PDF, Apple Computers
- Apple Dictionary, Edition Oxford american dictionary, word:suggestion, Version 1.0.0 (1.0.0), 2005
- Peter Blickford, Interface Design: The Art of Developing Easy-to-Use Software, Academic Press, 1997
- Alan Cooper, The inmates are running the asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity, SAMS Publishing, 2004
- Alan Cooper, 'The Myth of Metaphor', Visual Basic Programmer's Journal, June 1995, [http://www.cooper.com/articles/art_myth_of_metaphor.htm], 3.9.2005
- Alan Cooper&Robert Reimann, About Face 2.0: The essentials of interaction design, Wiley Publishing, Indianapolis 2003
- Robert X. Cringely, Triumph of the nerds: THE TELEVISION PROGRAM TRANSCRIPTS: PART III, N.D., [<http://www.pbs.org/nerds/part3.html>], 3.9.2005
- Encyclopedia Britannica Online, Graphical User Interface, N.D., [<http://www.britannica.com/ebc/article-9365993?query=Graphical%20User%20interface>], 3.9.2005
- Glenn R. Fong, ARPA Does Windows: The Defense Underpinning of the PC Revolution, PDF, Business and Politics 2001
- Scott Gasch, Alan Kay, Fall 1996, [<http://ei.cs.vt.edu/history/GASCH.KAY.HTML>], 3.9.2005
- François Guimbretière, Paper Augmented Digital Documents, 2003, PDF, Proceedings of the 16th annual ACM symposium on User interface software and technology, ACM Press
- Kenton O'Hara&Abigail Sellen, A Comparison of Reading Paper and On-Line Documents, 1997, pdf, Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press
- Alison J. Head, Personas: Setting the Stage for Building Usable Information Sites, Online Magazine Vol. 27 No. 4 – July/August 2003, [<http://www.infotoday.com/online/jul03/head.shtml>], 3.9.2005
- Andy Hertzfeld, I still remember Regions, April 1982, [http://www.folklore.org/StoryView.py?story=_Still_Remember_Regions.txt], 3.9.2005
- Bruce Horn, On Xerox, Apple and Progress, 1996, [http://www.folklore.org/StoryView.py?project=Macintosh&story=On_Xerox,_Apple_and_Progress.txt], 3.9.2005
- Catherine C. Marshall, Annotation: from paper books to the digital library, 1997, pdf, Proceedings of the second ACM international conference on Digital libraries, ACM Press
- Catherine C. Marshall&A.J. Bernheim Brush, Exploring the Relationship between Personal and Public Annotations, 2004, PDF, Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries, ACM Press

- Donald A. Norman, *The design of everyday things*[The psychology of everyday things], Basic Books, New York 1988

- Jef Raskin, *The Human Interface*, 2000, Addison-Wesley(german version), 2001, originally published by ACM Press 2000

- Bill N. Schilit, Morgan N. Price, Gene Golovchinsky, *Digital Library Information Appliances*, 1998, Proceedings of the third ACM conference on Digital libraries, ACM Press

- Abigail Sellen & Richard Harper, *Paper as an Analytic Resource for the Design of New Technologies*, 1997, PDF, Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press

- John Siracusa, *Metadata, the Mac, and you*, 20.8.2001, [<http://arstechnica.com/reviews/os/metadata.ars>], 3.9.2005

- John Siracusa, *About the finder...*, 2.4.2005, [<http://arstechnica.com/articles/paedia/finder.ars/2>], 3.9.2005

- Bruce Tognazzi, *AskTog—OSX: A First Look*, february 2000, [<http://www.asktog.com/columns/034OSX-FirstLook.html>], 3.9.2005

- Wikipedia, *Annotation*, 30.8.2005, [<http://en.wikipedia.org/w/index.php?title=Annotation&oldid=22168830>], 3.9.2005

- Wikipedia, *IMF*, 13.10.2005, [http://en.wikipedia.org/wiki/International_Monetary_Fund], 13.10.2005

- Wikipedia, *'CascadingStyleSheets'*, subchapter *'Use Of CSS'*, 12.10.2005, [http://en.wikipedia.org/wiki/Cascading_Style_Sheets], 12.10.2005

- Joanna L. Wolf, *Effects of Annotations on Student Readers and Writers*, 2000, PDF, Proceedings of the fifth ACM conference on Digital libraries, ACM Press

- Xerox PARC—Palo Alto Research Center, *About PARC*, 12.8.2003, [<http://www.parc.xerox.com/about/history/default.html>], 3.9.2005

FIGURES

C.2 This list displays the figures found in this thesis.

Categorized by chapter

■ Chapter 02: The evolution of the window

FIGURE 2.1 Memex
 FIGURE 2.2 Sketchpad
 FIGURE 2.3 Xerox Alto
 FIGURE 2.4 Document Icons (Mac OS 9 / Mac OS X / Windows)
 FIGURE 2.5 Microsoft Windows Dialog Box
 FIGURE 2.6 Apple Mac OS X Dialog Box
 FIGURE 2.7 Microsoft Windows Preference Window
 FIGURE 2.8 Apple Mac OS X Preference Window
 FIGURE 2.9 Microsoft Windows Document Window
 FIGURE 2.10 Apple Mac OS X Document Window
 FIGURE 2.11 Microsoft Windows Application Window
 FIGURE 2.12 Apple Mac OS X Application Window

■ Chapter 04: Reading mechanism on paper and online

FIGURE 4.1 An example of document placement layout during the planning phase of writing

■ Chapter 05: Relationship between paper and electronic documents

FIGURE 5.1 The activity profile for 8 desk officers and chiefs at the IMF.

■ Chapter 06: From personal to public annotations

FIGURE 6.1 WebAnn embedded in Microsoft Internet Explorer

■ Chapter 09: Reviewing the current annotation solutions

FIGURE 9.2.1 EditMode
 FIGURE 9.2.2 AnnotationMode
 FIGURE 9.2.3 IncorporationMode
 FIGURE 9.3.1 The PADD Infrastructure
 FIGURE 9.3.2 Annotation workflow
 FIGURE 9.4.1 XLibris
 FIGURE 9.5.1 Commenting toolbar
 FIGURE 9.5.2 Note inside a PDF
 FIGURE 9.5.3 TextEdits menu
 FIGURE 9.5.4 Highlights menu
 FIGURE 9.5.5 Attachments menu
 FIGURE 9.5.6 Markups toolbar
 FIGURE 9.5.7 Callout tool, Cloud tool, Arrow tool
 FIGURE 9.5.8 Authorlist

■ Chapter 10: Neon 1.0

FIGURE 10.1.1 The foil animation
 FIGURE 10.1.2 Apple Mac OS X vs. Microsoft Windows
 FIGURE 10.2.1 Annotation toolbar in the document window
 FIGURE 10.2.2 Inspiration sheet and first icon approaches
 FIGURE 10.2.3 Color selection process for the mouse input.
 FIGURE 10.2.4 Color selection process for the keyboard input.
 FIGURE 10.3.1 Drawer on the left side of the window
 FIGURE 10.3.2 Fontcard of the freefont Hybi
 FIGURE 10.4.1 Old menubar vs. new menubar
 FIGURE 10.4.2 Open menubar item
 FIGURE 10.4.3 Notification on icon level
 FIGURE 10.4.4 Notification on menubar level

■ Chapter 11: Goal-Directed design

FIGURE 11.1 Picture of John

FIGURE 11.2 Picture of Christoph

FIGURE 11.3 Picture of Thomas

■ Chapter 12: Neon 2.0

FIGURE 12.1.1 Seven stages of action by Donald A. Norman

FIGURE 12.1.2 Examples of Anchor-only annotations

FIGURE 12.1.3 Examples of Emphasizers

FIGURE 12.1.4 Examples of Connectors

FIGURE 12.1.5 Annotationstructure

FIGURE 12.2.1 Annotation toolbox

FIGURE 12.2.2 New toolbar on document window

FIGURE 12.2.3 Author Info

FIGURE 12.2.4 Optionspalette

FIGURE 12.2.5 Neon button in titlebar

■ Chapter 13: Scenarios

FIGURE 13.2.1 Standard annotation palette vs. customized annotation palette

FIGURE 13.2.2 Neon in Microsoft Word

FIGURE 13.2.3 Setting a Neonfoil as public

FIGURE 13.2.4 Icon notification

FIGURE 13.2.5 Annotated document as an email attachment

FIGURE 13.3.1 Layout in the Adobe Photoshop environment

FIGURE 13.3.2 Annotated layout with the freeform ink mode

FIGURE 13.4.1 PDF document that Thomas would like to annotate

FIGURE 13.4.2 Option palette which allows Thomas to invite his colleague

FIGURE 13.4.3 Split-screen of the annotation session

FIGURE 13.4.4 Annotated blog article on Thomas computer

FIGURE 13.4.5 Addition in the HTML source to attach the annotation foil

FIGURE 13.4.6 Thomas' annotated article on a colleagues machine

■ Appendix B: The user test

FIGURE B.1 Number code from the test results

FIGURE B.2 Example page displaying the questionnaire.

FIGURE B.3 Example page displaying the drawing section

TABLES

C.3 Most tables in this thesis are visual interpretations of existing illustrations inside the quoted studies. Categorized by chapter

■ Chapter 03: Form&Functions of annotation

TABLE 3.1 Characteristics of annotations written in the student books

TABLE 3.2 Mapping annotation forms into functions.

■ Chapter 06: From personal to public annotations

TABLE 6.2 The annotation type categories

TABLE 6.1 Examples of coding for personal annotations

TABLE 6.3 Overview of materials collected. Each week, three readings were assigned.

TABLE 6.4 Overview of personal annotation types

TABLE 6.5 Correspondences between personal annotations (on paper) and public annotations (online).

■ Chapter 07: Effect of annotations on students

TABLE 7.1 Description of the Passages and the Annotations they received in the two Evaluative Annotation Conditions.

TABLE 7.2 Overview of the passage sets between the two evaluative groups