

Anwendungsdesign zur Erleichterung kontinuierlichen Weiterlernens

Thomas Vogt

Diplomarbeit
im Fach Informatik

Universität Hamburg
Juni 2003

Betreuer:
Prof. Dr. Horst Oberquelle
Prof. Dr. Christopher Habel

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation und Aufgabenstellung dieser Arbeit	1
1.2. Aufbau dieser Arbeit	2
1.3. Danksagung	2
2. Grundlagen über das Erlernen von Softwareanwendungen	5
2.1. Der Umgang mit komplexen Systemen	5
2.2. Lernpsychologische Grundlagen	6
2.2.1. Behaviorismus	6
2.2.2. Kognitivismus	6
2.2.3. Konstruktivismus	7
2.2.4. Erlernen von Softwareanwendungen	7
2.2.5. Motivation	9
2.3. Die Wissenskurve	10
2.4. »Intuitive Benutzbarkeit«	11
2.5. Allgemeine Anforderungen an die Gebrauchstauglichkeit	12
2.5.1. Minimalismus	14
2.5.2. Adaptivität und Adaptierbarkeit	15
2.5.3. Unterscheidung von Neulingen und Experten	16
2.6. Rechtliche Vorgaben und Standards	16
2.7. Folgerungen für das Anwendungsdesign zur Erleichterung kontinuierlichen Weiterlernens	17
3. Vorhandene Lösungen zur Unterstützung von Anwendern	19
3.1. Klassifikation anwenderunterstützender Systeme	19
3.1.1. Gebräuchliche Modelle	19
3.1.2. Merkmale für eine neue Klassifikation	20
3.1.3. Übersichtstabelle	22
3.2. Online-Handbücher	24
3.3. Tutorien	26
3.4. Tipps beim Anwendungsstart	28
3.5. Intelligente Hilfe	28
3.6. Nutzer- und Expertenmodellierung	32

3.7.	Explorationsunterstützung	33
3.7.1.	Über-Sichten	34
3.7.2.	Neutralmodus	34
3.7.3.	Spieldaten	34
3.7.4.	Explorationskarten	35
3.8.	Newsgroups und FAQs	35
3.9.	Anthropomorphe Schnittstellen	36
4.	Ein neues Konzept zur Lernunterstützung: Zeitverzögerte Nutzungshinweise	43
4.1.	Konzept	43
4.1.1.	Verzögerte Explorationskarten	44
4.1.2.	Verzögerte intelligente Hilfe	44
4.2.	Prototypische Implementation: Der <i>maxblue</i> Lotse	44
4.2.1.	Explorationskarten in Popup-Fenstern	46
4.2.2.	Verzögerte Hilfe per E-Mails	48
4.3.	Evaluation des Lotsen	48
4.4.	Bewertung der Ergebnisse	56
5.	Zusammenfassung und Ausblick	59
6.	Anhang	61
6.1.	Screenshots des <i>maxblue</i> Lotsen	62
6.2.	Beispiel einer vom Lotsen versendeten E-Mail	69
6.2.1.	Text der E-Mail	69
6.2.2.	Anhang der E-Mail	70
6.3.	Auswertungsdaten	71
6.4.	Literaturverzeichnis	72

Tabellen- und Abbildungsverzeichnis

Abb. 2–1:	Beispielhafte Wissenskurven (nach Nielsen)	10
Abb. 2–2:	Microsoft Excel im Editier-Modus	12
Abb. 2–3:	Schließen einer Datei in Microsoft Excel	13
Tab. 3–1:	Lernkomponenten, Hilfe- und Assistenzsysteme	23
Abb. 3–2:	Microsoft Access-Hilfe mit eingeklappten Registerkarten	24
Abb. 3–3:	Microsoft Access-Hilfe mit ausgeklappten Registerkarten	25
Abb. 3–4:	Beispiel einer Photoshop-Anleitung von Russell Brown	27
Abb. 3–5:	Tipp beim Start des Versionskontrollsystems WinCvs	28
Abb. 3–6:	Microsoft Office-Assistent von 1997	29
Abb. 3–7:	Aisa, ein Assistent für die Konfiguration von smart Autos	37
Abb. 3–8:	Nomi, die virtuelle Repräsentantin der novomind AG	37
Abb. 3–9:	Oberfläche von Microsoft Bob	40
Abb. 4–1:	Homepage von <i>maxblue</i>	45
Abb. 4–2:	Ausschnitt des <i>maxblue</i> Navigationsmenüs mit Lesezeichen	46
Abb. 4–3:	Explorationskarte für die Lesezeichen von <i>maxblue</i>	47
Abb. 4–4:	Entwicklung der Nutzerzahlen und der Anzahl an Lesezeichen	49
Tab. 4–5:	Entwicklung der Nutzerzahlen und der Anzahl an Lesezeichen	49
Abb. 4–6:	Anteil der aktiven Nutzer mit Lesezeichen	50
Tab. 4–7:	Anteil der aktiven Nutzer mit Lesezeichen	50
Abb. 4–8:	Täglicher Zuwachs an Lesezeichen	51
Tab. 4–9:	Täglicher Zuwachs an Lesezeichen	51
Abb. 4–10:	Wirkung von Popups und E-Mails	51
Tab. 4–11:	Wirkung von Popups und E-Mails	52
Abb. 4–12:	Profil der E-Mail-Empfänger	52
Tab. 4–13:	Profil der E-Mail-Empfänger	52
Abb. 4–14:	Seitenaufrufe über Lesezeichen im Virtuellen Depot	53
Abb. 4–15:	Seitenaufrufe über Lesezeichen bei den Index-Übersichten	54
Abb. 4–16:	Seitenaufrufe über Lesezeichen im Aktien-Bereich	54
Abb. 4–17:	Seitenaufrufe über Lesezeichen auf anderen Detailseiten	55
Abb. 4–18:	Abnehmende Verwendung von Lesezeichen	55
Tab. 4–19:	Tatsächliche Verwendung von Lesezeichen (Übersicht)	56
Abb. 6–1:	Homepage von <i>maxblue</i> (komplett)	62
Abb. 6–2:	Anmelde-Seite von <i>maxblue</i>	63
Abb. 6–3:	»My maxblue« Startseite	64
Abb. 6–4:	Informationsseite des Lotsen	65
Abb. 6–5:	Spezielle Anmelde-Seite für den Lotsen	66
Abb. 6–6:	Einschalten der E-Mail-Option des Lotsen	67
Abb. 6–7:	Verwaltung der Einstellungen zum Lotsen	68
Abb. 6–8:	Angehängte PDF-Datei einer E-Mail des Lotsen	70
Tab. 6–9:	Tatsächliche Verwendung von Lesezeichen	71

1. Einleitung

»Ich brauche keinen Schachcomputer. Dass Maschinen den Menschen überlegen sind, kenne ich zum Beispiel schon von meinem Videorecorder.«
Harald Schmidt in seiner Show am 14. Mai 1997

1.1. Motivation und Aufgabenstellung dieser Arbeit

Softwareanwendungen sind komplexe Systeme. Programme wie etwa zur Bild- oder Textverarbeitung bieten den Nutzern eine unüberschaubare Fülle von Funktionalitäten, von denen die Mehrheit für einen einzelnen Nutzer und seine Aufgabenerfordernisse jeweils irrelevant ist. Zu einem Teil rührt die hohe Komplexität von dem marktgetriebenen Trend her, dass Hersteller alle möglichen Funktionen in einer einzigen Anwendung bündeln [Encarnaçao 97]. Viele Softwarehersteller versuchen, sich von ihren Wettbewerbern vor allem dadurch zu differenzieren, dass sie in ihren Produkten neue und noch mehr Funktionen als die Konkurrenz implementieren. Dies gilt übrigens nicht nur für Computeranwendungen: Wegen Systemen, die mit Funktionen völlig überladen sind, ist zu hohe Komplexität auch im Handy-Bereich ein Problem für die Nutzer [Heise 2003].

Die nicht zu beherrschende Komplexität folgt aber auch daraus, dass Anforderungen an die Gebrauchstauglichkeit zu wenig berücksichtigt werden. Im Vordergrund steht eben die Bereitstellung möglichst vieler Funktionen. Die Bedürfnisse der Nutzer und die Anforderungen an die Gebrauchstauglichkeit werden von den Herstellern nicht ausreichend beachtet, was die Bedienbarkeit am meisten erschwert [Heise 2002]. Selbst erfahrene Nutzer können daher die Leistungsfähigkeit eines Systems nicht voll ausreizen. Ich beschäftige mich deswegen mit dem Thema, wie Nutzer, die ein System bereits kennen, das angebotene Potential besser ausschöpfen können.

Es ist bekannt, dass in der anfänglichen Einarbeitungszeit am meisten über ein neues System gelernt wird. Hat man sich erst mal zurecht gefunden, lernt man bei der weiteren Nutzung nur noch wenig Neues über das Programm und dessen Bedienung hinzu. Die aus der Einarbeitungszeit gewonnenen Erkenntnisse werden später kaum vertieft oder ausgebaut, so dass die Nutzer nur den gewohnten, jedoch beschränkten Funktionsumfang verwenden. Die Lernkurve, oder besser gesagt die Wissenskurve, steigt am Anfang also zunächst an, um dann jedoch relativ schnell wieder abzuflachen. (Auf den missverständlichen Begriff »Lernkurve« und die in diesem Zusammenhang passendere Bezeichnung »Wissenskurve« gehe ich im nächsten Kapitel näher ein.) Im Web-Bereich lässt sich diese Erfahrung auf Sites, die für wiederkehrende Besucher ausgelegt und hinreichend komplex sind,

übertragen. Als Beispiele seien hier nur Online-Banking-Anwendungen und Webmail-Programme aufgeführt.

Nutzern bleibt so ein großer Teil der im System enthaltenen Funktionalität verborgen. Bei Systemen mit großer Komplexität hat dies zur Folge, dass ein einzelner Nutzer, selbst wenn er bereits lange mit einem System arbeitet, nicht alle für ihn relevanten Funktionalitäten kennen lernt. Dieses Unkenntnis hat eine reduzierte Effektivität und Effizienz zur Folge [Linton et al. 2000]. Viele Aufgaben könnten Nutzer besser und schneller bewältigen, wenn sie ihre Kenntnisse über das System nach der Einarbeitungsphase stärker ausbauen. Deshalb wäre es wünschenswert, wenn den Nutzern vom System nicht nur ein leichter Einstieg geboten wird, sondern sie auch in ihrem alltäglichen Umgang dabei unterstützt werden, kontinuierlich mehr über das System zu lernen.

In dieser Arbeit untersuche ich daher, was unternommen werden kann, um dieses kontinuierliche Weiterlernen für den Nutzer zu erleichtern. Kann man es schaffen, wieder eine kleine Steigung in der sonst abgeflachten Wissenskurve anzuregen? Es werden Ideen und Konzepte für das Anwendungsdesign entwickelt, welche die Nutzer stärker bei dem Lernprozess während der alltäglichen Arbeit mit dem System unterstützen. Ich verwende dabei den Begriff Weiterlernen, um den Unterschied zur anfänglichen Lernphase beim ersten Umgang mit der Anwendung zu verdeutlichen.

1.2. Aufbau dieser Arbeit

Zunächst setze ich mich mit dem Aspekt des Erlernens von Softwareanwendungen auseinander. Den Einstieg bildet eine nähere Beschreibung des Problemfeldes. Danach wende ich mich den lernpsychologischen Grundlagen zu. Bei diesem äußerst umfangreichen Thema beschränke ich mich auf die wesentlichen Aspekte, die für diese Arbeit bedeutsam sind. Ich beschreibe anschließend Wissenskurven, betrachte den Begriff der »intuitiven Benutzbarkeit« und stelle allgemeine Anforderungen an die Gebrauchstauglichkeit eines Systems dar. Wie beim Abschnitt über die lernpsychologischen Grundlagen beschränke ich mich auch beim letztgenannten Punkt auf die Aspekte mit Bezug zur Erlernbarkeit. Im Anschluss beschäftige ich mich mit rechtlichen Vorgaben und Standards wie beispielsweise den entsprechenden ISO-Normen, bevor ich die Ergebnisse als Anforderungen zur Erleichterung kontinuierlichen Weiterlernens zusammenfasse.

Nach diesen eher abstrakten und theoretischen Betrachtungen analysiere ich im nächsten Kapitel vorhandene Lösungen aus der Praxis, die das Erlernen von Softwareanwendungen erleichtern sollen. Dabei orientiere ich mich an einer eigenen Klassifikation von Ansätzen, die vor allem Gestaltungsmerkmale betont. Zu den behandelten Ansätzen gehören u.a. klassische Online-Handbücher, Tutorien, das Einblenden von Tipps beim Anwendungsstart und intelligente Hilfesysteme. Besonders interessant ist dabei die Betrachtung der Entwicklungsgeschichte des »berühmt-berüchtigten« Microsoft Office-Assistenten.

In einem weiteren Kapitel entwickle ich eigene, neue Lösungsansätze. Dabei handelt es sich um Ansätze, bei denen Nutzungshinweise verzögert gegenüber der Aufgabenerledigung ausgegeben werden. In Zusammenarbeit mit der Firma SinnerSchrader wurden diese Ideen an einem Fallbeispiel aus dem Web-Bereich erprobt. Die Ansätze wurden für die deutsche Version des europäischen Online-Brokers *maxblue* (<http://www.maxblue.de/>) umgesetzt.

Den Abschluss der Arbeit bildet eine Zusammenfassung, in der die umgesetzten Lösungsideen bewertet werden.

1.3. Danksagung

Für die Betreuung meiner Diplomarbeit bedanke ich mich an erster Stelle bei Horst Oberquelle und Christopher Habel vom Fachbereich Informatik der Universität Hamburg. Nicole Krämer vom Psychologischen Institut der Universität zu Köln hat mich bei der Bearbeitung des Themas

»Anthropomorphe Schnittstellen« unterstützt. Paolo Petta vom Austrian Research Institute for Artificial Intelligence half mir mit der Bereitstellung des IntelliSense-Whitepapers von Microsoft.

Außerordentlicher Dank gilt der Firma SinnerSchrader, die mir die Gelegenheit gegeben hat, die Arbeit in einem idealen Umfeld durchzuführen. Mit viel Engagement haben Stefan Kunze, Holger Blank, Christian Jung, Moritz Koch, Detlef Wichmann, Michael Roth, Michael Boos, Stephan Groneberg, WenLing Tan, Nicola Beck, Gunnar Schmidt, Dirk Lehmannski, Frank Nögler, Jorge Oliveira, Thorsten Thomsen, Christoph Trautwein, Markus Reinedahl, Andreas Steffan, Anke Becker, Kai Thomsen und Steffen Becker zum Gelingen beigetragen. Das hat wirklich Spaß gemacht!

Ebenso gilt ein großer Dank Alexander Walter, Markus Pfister und Melanie Voß von der Deutschen Bank, die mir die Umsetzung der entwickelten Ideen bei *maxblue* ermöglichten.

Eine unersetzbare Hilfe ist meine Frau Hakima.

2. Grundlagen über das Erlernen von Softwareanwendungen

»Die einzig vernünftige Lösung ist, die Handbücher auseinanderzunehmen, sie sechs Monate lang mit Hilfe eines Etruskologen zu studieren sie auf ein paar Karteikärtchen zu komprimieren (was völlig reicht) und sie dann wegzuwerfen.« [Eco 85, S. 47f]
Umberto Eco über Benutzerhandbücher für Personalcomputer

Zu Beginn dieses Kapitels beschreibe ich das Problemfeld dieser Arbeit näher. Anschließend erläutere ich die lernpsychologischen Grundlagen, wobei ich kurz auf die wesentlichen Lerntheorien des Behaviorismus, Kognitivismus und Konstruktivismus eingehe und Motivationsaspekte eingehender beschreibe. Ich kläre die Verwendung der Begriffe »Lernkurve« und »Wissenskurve« und beschreibe typische Verläufe. Ebenso wird der oft fälschlich verwendete Ausdruck »intuitive Benutzbarkeit« betrachtet. Bei den allgemeinen Anforderungen an die Gebrauchstauglichkeit gehe ich näher auf minimalistische Unterstützungsansätze ein und beschäftige mich mit den Themen der Adaptivität und Adaptierbarkeit von Softwareanwendungen, sowie der Unterteilung von Nutzern in Neulinge und Experten. Anschließend sammle ich die Forderungen, die Gesetze und Normen für das Erlernen von Softwareanwendungen aufstellen. Den Kapitelabschluss bildet eine Zusammenfassung der Ergebnisse als Folgerungen für das Anwendungsdesign zur Erleichterung kontinuierlichen Weiterlernens.

2.1. Der Umgang mit komplexen Systemen

Typischerweise erweitern Nutzer ihre Kenntnisse über komplexe Systeme in ihrem alltäglichen Umgang nur äußerst wenig. Sie hören auf, mehr über eine Benutzungsschnittstelle zu lernen, sobald sie genug wissen, um die grundlegenden Dinge zu erledigen. [Nielsen 2002] beschreibt diese Tatsache mit dem Ausdruck »asymptoting user performance«. Damit soll deutlich werden, dass eine anfangs steile Wissenskurve nach kurzer Zeit abflacht und kaum weiter steigt.

Dies Verhalten wird auch von anderen Autoren beschrieben (z.B. [Brodbeck/Rupietta 94], [Oppermann 94], [Houser/Deloach 98], [Bartz 2001]). Nutzer wenden sich an eine Anwendung, um eine Aufgabe auszuführen. Dies ist der Grund dafür, warum sie so schnell wie möglich mit den Anwendungsteilen arbeiten, deren Grundlagen sie beherrschen, und mit der Arbeit eben nicht so lange warten, bis sie alle Funktionen eines Anwendungssystems erlernt haben.

In einer Studie über die Wünsche der Nutzer von Microsofts Office-Anwendungen wurde festgestellt, dass Nutzer oft Funktionalitäten angefragt haben, die in den Produkten bereits implementiert waren. Vorhandene Funktionalitäten konnten viele Bedürfnisse der Nutzer eigentlich befriedigen, wurden aber nicht verwendet, weil sie unbekannt oder schwer zu finden waren. Die Nutzer konnten nicht alle Vorteile aus den Fähigkeiten der Software ziehen. Viele dieser Probleme wurden von einem lokalen Experten gelöst [Microsoft 96]. Ein lokaler Experte ist eine Person, die sich mit der betreffenden Softwareanwendung sehr gut auskennt – typischerweise jemand, der viel Erfahrung mit Computersystemen hat. Ein lokaler Experte wird oft als die beste Unterstützungsform für einen Nutzer angesehen (z.B. [Nielsen 93a], [Houser/DeLoach 98], [Kearsley 98], [Schaumburg 2001]). Microsoft hat daher versucht, einen lokalen Experten in der Software nachzubilden. Auf den so entstandenen Microsoft Office-Assistenten und seine Entwicklungsgeschichte wird im Abschnitt über intelligente Hilfe eingegangen (Abschnitt 3.5, S. 29ff).

Es ist nicht so, dass die Lernphase über den Umgang mit einer Anwendung nach der Einarbeitung in die Grundlagen komplett abgeschlossen ist. Später lernt der Nutzer durchaus die Verwendung weiterer Funktionalitäten. Dies geschieht jedoch meist langsam und mühsam durch Versuch und Irrtum oder wie beschrieben im persönlichen Kontakt mit Kollegen, Bekannten oder anderen Personen [Linton et al. 2000].

Die Stagnation der Wissenskurve ist problematisch, weil Nutzer durch das Anwenden weiterer, sonst ungenutzter Funktionalität oft mehr Zeit sparen könnten, als sie für das Erlernen der entsprechenden Funktionalität aufbringen müssten [Nielsen 93a]. Damit die Nutzer ihr Wissen über ein System einfacher erweitern können, benötigen sie eine Unterstützung, die parallel zu ihrer Aufgabebearbeitung gegeben wird. Dies ist ein wesentlicher Unterschied zu klassischen Lernsituationen, in denen reine Lernsysteme eingesetzt werden können.

2.2. Lernpsychologische Grundlagen

Als Basis für die weiteren Betrachtungen werden als erstes die drei grundlegenden lernpsychologischen Konzepte des Behaviorismus, Kognitivismus und Konstruktivismus beleuchtet, um einen kurzen Überblick über diese zu geben.

2.2.1. Behaviorismus

Der Behaviorismus betrachtet Lernen als Steuerung durch Reize, um unerwünschtes Verhalten durch Bestrafung zu vermeiden und erwünschtes Verhalten durch Belohnung zu verstärken, ohne dabei näher auf den Lernenden einzugehen. Neue Verhaltensmuster werden so lange wiederholt, bis sie automatisiert worden sind. Nach der Reinforcement-Theorie von Skinner zeigen Menschen am ehesten gewünschtes Verhalten, wenn sie hinterher sofort dafür gelobt werden. Geeignet ist die behavioristische Sichtweise nur für einfache Problemstellungen oder zur reinen Vermittlung von Faktenwissen, wie z.B. beim Vokabellernen (vgl. [Donker 2002], [Mergel 98]). Für so eine komplexe Aufgabe, wie dem Ausbauen des Wissens über eine Softwareanwendung ist dieser Ansatz nicht geeignet.

2.2.2. Kognitivismus

Anders als beim Behaviorismus stehen beim Kognitivismus der Lernende und der Lernprozess im Mittelpunkt. Lernen wird als Informationsverarbeitungsprozess betrachtet und das Gehirn als informationsverarbeitendes Gerät, das Eingaben zu Ausgaben verarbeitet, etwa analog einem Computer. Beim Lernen wird Wissen aufgenommen und z.B. durch Aufbau mentaler Modelle verarbeitet. Die Speicherung des neuen Wissens geschieht durch Anpassungen kognitiver Strukturen und Prozesse. Um zu lernen, müssen die neuen Informationen verständlich dargestellt werden, und der Lernende muss die geeigneten kognitiven Operationen ausführen. Verschiedene Informationsarten werden nach der kognitivistischen Sichtweise unterschiedlich verarbeitet und gespeichert. Es wird dazu zwischen deklarativem, prozeduralem und kontextuellem Wissen unterschieden. Deklaratives

Wissen ist die Kenntnis von Sachverhalten, prozedurales Wissen ist Information über Fertigkeiten und kontextuelles Wissen bezieht sich auf situative, fallbezogene Kenntnisse [Donker 2002].

2.2.3. Konstruktivismus

Ähnlich wie beim Kognitivismus liegt auch beim Konstruktivismus der Schwerpunkt bei dem Lernenden und seinen Verstehensprozessen. Der Konstruktivismus betont dabei allerdings viel stärker individuelle Unterschiede. Es gibt nicht nur einen Informationsverarbeitungsprozess wie beim Kognitivismus, sondern das Wissen wird von jedem Lernenden, abhängig von seinem Vorwissen, seinen Erfahrungen und seinen Plänen, aktiv konstruiert. Die Unterschiede an Vorkenntnissen, Erfahrungen und Plänen lassen Lernende dieselben Dinge unterschiedlich wahrnehmen und interpretieren, so dass Wissen immer ganz individuell konstruiert wird. Der Lernende muss die vermittelten Lerninhalte aktiv bearbeiten, um sie in eine Form zu bringen, in denen sie zu seinem Vorwissen passen und seinen Kenntnissen hinzugefügt werden können. Die jeweiligen Lernwege sind daher auch nicht vorhersehbar. Wissen wird als nicht vermittelbar angesehen und Lehren kann nur eine Unterstützung und Anregung bei der aktiven Wissenskonstruktion sein [Donker 2002].

2.2.4. Erlernen von Softwareanwendungen

Moderne Computersysteme haben einen sehr hohen Komplexitätsgrad und in typischen, alltäglichen Anwendungssituationen müssen sich die Nutzer alleine und selbstständig mit der Software auseinandersetzen. Zu solchen komplexen und selbstständig zu bewältigenden Situationen passt der konstruktivistische Ansatz [Schudnagis/Womser-Hacker 2002]. Er ist daher am besten geeignet, den Prozess beim Erlernen des Umgangs mit einer Softwareanwendung zu beschreiben. Verschiedene Aspekte dieses Prozesses sollen nun näher betrachtet werden.

Lernen erfordert Umgebungen, in denen Nutzer aktive Mitwirkende statt nur passive Konsumenten sein können. Um große Lernerfolge zu erzielen, sollte man die Nutzer mit dem arbeiten lassen, was sie lernen. Dazu gibt es mehrere Möglichkeiten (vgl. [Fischer/Scharff 98]): Man kann zwei Verfahren für die Bearbeitung einer Aufgabe beschreiben und die Nutzer dann entscheiden lassen, welches in einer vorgegebenen Situation das richtige ist. Man kann die Nutzer auch dazu auffordern, darüber nachzudenken, wie die gelernten Verfahren auf ihre eigene Arbeit anzuwenden sind. Dies ist das Prinzip, auf dem Tutorien basieren (siehe Abschnitt 3.3 Tutorien, S. 26ff). Man kann ihnen kurz Fakten erklären und sie später danach fragen, sich an das Gelernte zu erinnern. Wichtig ist es, etwas mehr zu tun, als bloß Schritt-für-Schritt die Verfahren zu präsentieren und zu erwarten, dass diese auswendig gelernt werden.

Die Aktivität der Nutzer entspricht dabei auch ganz dem normalen Verhalten [Carroll/Aaronson 88]. Bei einer typischen Tätigkeit (wie Arbeiten oder Spielen) sind Individuen von sich aus solange aktiv, bis sie auf eine Störung stoßen und sie über die Störung nachdenken. Diese Störungen, die von fehlendem Wissen, Missverständnissen über die Auswirkungen von Handlungen usw. herrühren, sind sehr gute Gelegenheiten zum Lernen [Fischer/Scharff 98]. Intelligente Hilfesysteme versuchen, diese Situationen zu erkennen und für ihre automatische Unterstützung zu nutzen (siehe Abschnitt 3.5 Intelligente Hilfe, S. 28ff).

Das aktive Konstruieren von Wissen beschäftigt die Nutzer und berücksichtigt ihr Bedürfnis, das Wissen auf eine für sie persönlich geeignete Weise zu konstruieren. Das bestehende Wissen wird dazu genutzt, um neues Wissen zu konstruieren. Menschen lernen neue Information in Begriffen ihres aktuellen Verständnisses [Sebrechts/Marsh/Furstenburg 90]. Menschen haben einmalige, persönliche Hintergründe, vor denen sie aktiv werden und die ihre Erwartungen und Ziele formen. Ihre persönlichen Ziele und Überlegungen richten ihre Aufmerksamkeit nur auf eine Teilmenge der Objekte und Ereignisse, die sie potentiell wahrnehmen könnten. Sie tendieren dazu, Dinge nicht wahrzunehmen, die nach ihrer aktuellen Analyse der Situation irrelevant sind [Carroll/Aaronson 88]. Um ein möglichst breites Spektrum an bestehendem Wissen zu erfassen, müssen Umgebungen auf die Nutzer zugeschnittene Informationen bieten, indem sie z.B. unterschiedliche Beschreibungen neuer Informationen enthalten [Fischer/Scharff 98].

Eine weitere Möglichkeit die Nutzer zu aktivieren, ist die Aufforderung an sie, selbständig herauszufinden, wie sich das System verhält. Allgemein wird dieser Ansatz entdeckendes Lernen genannt (siehe dazu [Donker 2002] und [Lewis/Rieman 93]). Im Zusammenhang mit dem Erlernen des Umgangs mit einer Softwareanwendung ist die Bezeichnung »exploratives Lernen«, an die ich mich in dieser Arbeit halte, ebenfalls sehr verbreitet. Bei dieser Lernform werden die relevanten Informationen nicht fertig strukturiert präsentiert, sondern der Lernende muss Informationen finden, priorisieren und ordnen, um Nutzen aus ihnen zu ziehen. »Die Exploration wird geleitet von Neugier und Interesse des Lernenden. Er soll Lösungen für interessante Fragen entwickeln, statt Fakten auswendig zu lernen. Der Stellenwert des impliziten Lernens und der Intuition wird ebenfalls betont. Ziel des Lernens ist die Ausbildung der Problemlösefähigkeit.« [Donker 2002, S. 156] Bei der explorativen Interaktion mit Softwareanwendungen erforscht der Nutzer die Benutzungsoberfläche, die Funktionalität des Systems oder die aktuelle Arbeitsaufgabe. Dabei wird der jeweilige Gegenstand nicht vollständig, sondern jederzeit nach Bedarf erforscht. Diese Art der Interaktion wird in zwei Formen eingeteilt: In erkundende und experimentierende Nutzung. Mittels Erkundung verschafft sich der Nutzer einen Überblick über die Systemeigenschaften und sammelt Erfahrungen mit der Anwendung. Er betrachtet das System analysierend, sucht für seine aktuellen Ziele passende Funktionen und stellt Hypothesen zum Systemverhalten auf. Diese Hypothesen werden anschließend durch Experimentieren aktiv überprüft, indem die gefundenen Funktionen erprobt werden. Aufgaben werden durch neue Kombinationen bekannter oder Einsatz bisher unbekannter Funktionen vom Nutzer selbständig gelöst, bzw. ihre Bearbeitung optimiert. Die Vorteile explorativen Lernens sind die Schulung von Hypothesenbildung und die Förderung von Kreativität, von Transferleistungen und von intrinsischer Motivation (zu Motivation siehe den nächsten Abschnitt, S. 9f). Dieser Qualifizierungsprozess richtet sich an den Arbeitsaufgaben der Nutzer aus und wird permanent durchlaufen [Paul 93]. Daher beschreibt er nicht nur die anfängliche Einarbeitungszeit von Neulingen, sondern auch spätere Nutzungs- und Lernphasen erfahrener Nutzer. Auch die versuchen als erstes, selbständig durch Exploration oder Experimente aus einer Sackgasse in der Benutzungsschnittstelle heraus zu kommen, bevor sie als nächstes eine andere Person, wie etwa einen lokalen Experten, um Unterstützung bitten, oder sich zuletzt doch noch an eine Online-Hilfe wenden [Apple 2002].

[Paul 93] leitet seine Ideen zur explorationsfreundlichen Gestaltung von Computersystemen aus den Erfahrungen mit Computerspezialisten ab. Menschen, die seit langer Zeit mit Computern arbeiten, haben sich durch ihren reichen Erfahrungsschatz experimentierende und erkundende Umgangsweisen mit Softwaresystemen angeeignet. Sein Ziel war es, die Vorteile dieser Methoden auch für Menschen mit weniger Computererfahrung nutzbar zu machen. Für explorative Umgangsweisen betrachte ich nun, ob sie für Männer wie für Frauen gleichermaßen passend sind und für welche Typen von Softwareanwendungen sie geeignet sind.

Es gibt viele Studien zu geschlechtsspezifischen Unterschieden beim Umgang mit Computern. Solche Unterschiede zeigen sich beispielsweise in computerbezogenen Kompetenzen, in der tatsächlichen Nutzung von Computern und in Einstellungen gegenüber Computern. Häufig wird festgestellt, dass Männer umfassendere theoretische und praktische Kenntnisse über Computer besitzen als Frauen, im Umgang mit dem Computer vertrauter und sicherer sind, positivere Einstellungen gegenüber Computern haben und Computer bereits seit längerer Zeit und intensiver nutzen. Die Studien geben jedoch durchaus kein eindeutiges Bild ab. Die Ergebnisse der Studien sind teilweise inkonsistent und es konnten manchmal gar keine geschlechtsspezifischen Effekte nachgewiesen werden. Außerdem deuten einige Ergebnisse an, dass sich die Unterschiede in den letzten Jahren verändert haben. Es gibt offensichtlich Unterschiede, aber ein großer Teil davon kann auf die unterschiedliche Zeit, seit der Computer genutzt werden, zurückgeführt werden. Der derzeitige Stand der Forschung ermöglicht es jedenfalls nicht, Empfehlungen zu unterschiedlichen Unterstützungsarten für Frauen und Männer zu geben [Richter/Naumann/Horz 2001]. Daher wird der explorative Ansatz als geeignet für beide Geschlechter angenommen.

Ein explorativer Umgang ist für interaktive Systeme geeignet, die nutzergeführt sind. Bei diesen Systemen entscheiden die Nutzer, wie sich der Arbeitsablauf gestaltet und welche Funktionalitäten sie dabei einsetzen. Das System gibt keine bestimmte Auswahl oder feste Abfolge vor, so dass die Nutzer sehr viele Freiräume haben [Paul 93]. Dazu können Anwendungen gezählt werden, bei denen Nutzer direkt-manipulativ vorgehen oder die das WYSIWYG-Prinzip umsetzen. Die Nutzer müssen auf ihre

Eingaben unmittelbare Reaktionen bekommen, damit sie die Auswirkungen ihrer explorativen Versuche sofort einschätzen können.

Exploration ist ungeeignet bei Anwendungen, für die eine systemgeführte Interaktion kennzeichnend ist, in der die Software die Auswahl der Funktionen und die Reihenfolge der Arbeitsschritte festlegt. Bei solchen Systemen reduziert sich die Nutzung auf das Auswählen von vorgegebenen Ablaufvariationen und die Eingabe vorgangsspezifischer Daten. Beispiele dafür sind Steuerungsanlagen für komplexe, technische Anlagen wie Großraumflugzeuge, Kraftwerke oder Pipelines [Paul 93]. Auch jede Art von Batchprozessen und Systeme, in denen Nutzer erst relativ spät die Auswirkungen ihrer Handlungen sehen, sind für Exploration nicht geeignet.

Unabhängig von der Anwendung sind auch nicht alle Aufgaben geeignet, explorativ erarbeitet zu werden. Bei sehr einfachen Aufgaben, z.B. dem Einlegen von neuem Papier in ein Faxgerät, haben Nutzer vielleicht gar nicht die Geduld zum explorativen Lernen [Redish 98].

Mit Rücksicht auf diese Einschränkungen eignet sich exploratives Lernen für eine Erleichterung kontinuierlichen Weiterlernens bei nutzergeführten Softwareanwendungen, wenn die Nutzer hinreichend komplexe Aufgaben bearbeiten, wie sie im normalen Anwendungsalltag am häufigsten auftreten.

2.2.5. Motivation

Beim Lernen sind Motivationsaspekte ebenso wichtig wie die bisher betrachteten Kognitionsaspekte [Fischer/Scharff 98]. Hinweise helfen nur dann, wenn die Nutzer zum Lernen und zur Arbeit motiviert sind. Diese Motivation sollten die Hinweise selbst wecken und erhalten [Mirel 98].

Bei Motivation wird zwischen intrinsischer oder innerer und extrinsischer oder äußerer unterschieden (vgl. [Hochhaus o.J.] und das Glossar in [Rühl 99]). Intrinsische Motivation beschreibt allgemeine Neugier oder direktes Interesse am Thema, während bei extrinsischer Motivation das Hauptinteresse außerhalb des Themas liegt. Im Zusammenhang mit dem Erlernen von Softwareanwendungen spricht man z.B. dann von intrinsischer Motivation, wenn jemand den besseren Umgang mit der Software selbst erlernen will, und von extrinsischer, wenn die konkrete Softwareanwendung nur eine Nebenrolle spielt und jemand primär an beruflichem Aufstieg oder der Erlangung eines Zertifikats interessiert ist. In letzterem Fall ist das Erlernen dann nur ein Mittel für einen anderen Zweck, der im Vordergrund steht. Intrinsische Motivation ist tragfähiger und dauerhafter als extrinsische, und bei intrinsischer Motivation ist die Bereitschaft zum Lernen höher. Daher soll untersucht werden, wie eine Steigerung v.a. der intrinsischen Motivation angeregt werden kann. [Rühl 99] zählt dafür mehrere Möglichkeiten auf:

- aktive Teilnahme des Nutzers am Lernprozess (physisch und/oder mental). Aktives Lernen steigert bereits die Motivation und erhöht auch die Fähigkeit, das Gelernte später anzuwenden, macht den Lernprozess aber auch anspruchsvoller [Rosson/Carroll 2002].
- Spaß: humorvolle, überraschende Elemente können Interesse wecken und steuern. [Carroll 96] hat beispielsweise gute Erfahrungen damit gemacht, als Inhalt eines Tutoriums zum Erlernen einer Programmiersprache die Analyse eines Programms für ein Glücksspiel zu wählen. Humor kann aber auch übertrieben und lästig wirken, zumal das Humorverständnis stark kulturell geprägt ist.
- Abwechslung: ein möglichst vielfältiges Angebot unterschiedlicher Medien und Tätigkeiten ist empfehlenswert. Auch Bilder sind dazu geeignet, die Motivation zu erhöhen, da sie die Aufmerksamkeit auf sich ziehen und so zur längeren und weiteren Beschäftigung verleiten. Sie erhöhen die Freude am Lernen, aktivieren Einstellungen, lösen Gefühle aus und animieren zum Lesen [Deimann 2002]. Eine erhöhte Bereitschaft zur längeren Beschäftigung mit einem Programm wurde auch bei der Anwesenheit eines virtuellen Helfers festgestellt [Krämer/Nitschke 2002] (siehe dazu Abschnitt 3.9 Anthropomorphe Schnittstellen, S. 36ff).
- Auswahl: der Lernende sollte seine persönlichen Lernwege selbst wählen können.

- soziale Interaktion: z.B. in Form von Teamarbeit, Gruppendiskussionen u.ä. Im Kontext von Softwareanwendungen liegt eine Einbindung von Diskussionsforen wie Newsgroups nahe (siehe Abschnitt 3.8 Newsgroups und FAQs, S. 35f).
- Fehlertoleranz: Fehler sind ein wichtiger Faktor beim Lernen. Deshalb wird empfohlen, eine ›sichere‹ Lernumgebung zu schaffen, in der keine demoralisierende Bestrafung zu erwarten ist.
- Erfolgsmessung und Anerkennung: möglichst mit positivem Maß. Bei den Rückmeldungen an den Nutzer sollte man sich auf Vorschläge zur Verbesserung konzentrieren statt auf die Fehler. Frühe Erfolgserlebnisse erzeugen Vertrauen und vermitteln ein positives Gefühl von Kompetenz [Frith 97].
- Herausforderung: keine zu trivialen, sondern herausfordernde Aufgaben. Besonders empfehlenswert sind selbst gesetzte Ziele der Lernenden. Um Neugier zu wecken, müssen Anreize geboten werden, die neu sind, aber nicht zu unterschiedlich von bereits Bekanntem. Total fremde Anreize können eher Angst als Neugier erzeugen. Es muss eine Balance zwischen Komplexität und Klarheit bzw. Übersichtlichkeit geben [Frith 97].

All diese Maßnahmen dienen der Förderung von für den Nutzer positiven Lernerfahrungen. Eine Verstärkung der Motivation erreicht man am ehesten, wenn man mehrere dieser Motive miteinander verknüpft.

Um eine vorhandene Lernmotivation aufrecht zu erhalten, muss der Sinn des Lernens so früh wie möglich vermittelt werden. Dazu müssen gleich zu Beginn einer Lektion die Ziele klar formuliert und Teilziele gesetzt werden [Hochhaus o.J.].

2.3. Die Wissenskurve

Bei Beschreibungen, wie Nutzer den Umgang mit einer Softwareanwendung erlernen, werden zur Illustration oft Kurven angegeben, auf denen der Kenntnis- bzw. Wissensstand der Nutzer über einer Zeitachse aufgetragen ist.

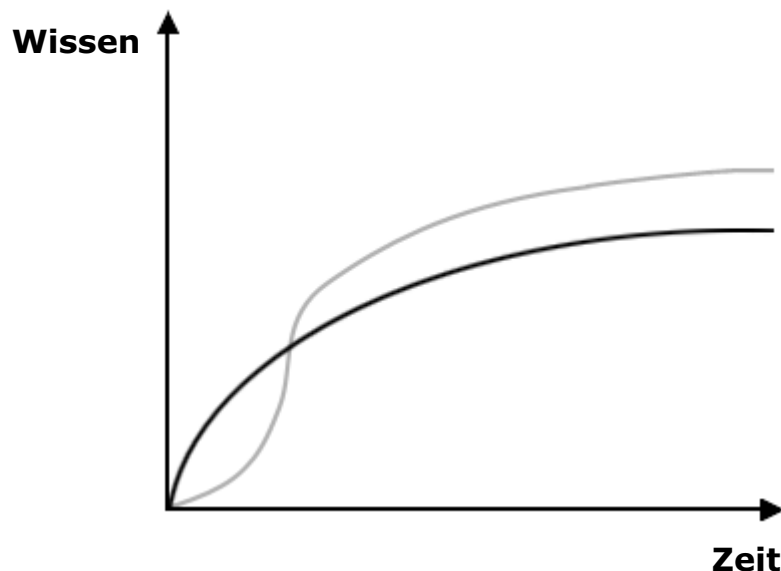


Abb. 2–1: Beispielhafte Wissenskurven (nach Nielsen)

Abbildung 2–1 ist eine Darstellung aus [Nielsen 93a, S. 28] nachgebildet, in der die Ordinate nicht nur die Kenntnisse über die Anwendung, sondern gleichzeitig auch die Effizienz der Nutzer beschreibt. Bei der dunklen Kurve hat der Nutzer in der Einarbeitungszeit von Beginn an einen großen Lernerfolg. Bei der hellen Kurve ist der Lernaufwand am Anfang hingegen recht hoch, was sich durch eine höhere Effizienz bezahlt macht, nachdem der Nutzer sich in das System (z.B. ein Kommandozeilensystem) eingearbeitet hat.

Meistens werden diese Kurven »Lernkurven« genannt, obwohl dies eigentlich falsch ist. Denn auf der Ordinate wird ja nicht aufgetragen, wie viel der Nutzer zu einem bestimmten Zeitpunkt lernt. In diesem Falle näherte sich die Kurve von oben dem Nullpunkt, um auf ihn zu stoßen, sobald der Nutzer nichts Neues mehr über die Anwendung dazulernt. Da jedoch aufgetragen wird, wie viel der Nutzer bis zu einem bestimmten Zeitpunkt bereits gelernt hat, symbolisiert der Wert auf der Ordinate die Kenntnisse oder das Wissen, das sich der Nutzer über die Anwendung angeeignet hat. Der Ausdruck »Wissenskurve« ist daher treffender.

Während es bei verschiedenen Kurven wie z.B. bei Abbildung 2–1 noch Unterschiede in der anfänglichen Lernphase gibt, ist allen das anschließende Abflachen gemeinsam. Wenn Nutzer dieses Plateau erreicht haben gelten sie als erfahren und bearbeiten ihre Aufgaben mit dem bis dahin erarbeiteten Kenntnisstand, ohne weitere, deutliche Lernfortschritte zu machen [Nielsen 93a]. Diese Plateaus sind sehr wichtig, kennzeichnen sie doch eine Phase, in der geeignete Denkstrukturen gebildet werden, damit sich die neuen Erkenntnisse und das gelernte Wissen festigen [Microsoft 2001].

Eine einfache Funktion, die in der Kognitionspsychologie zur Untersuchung von Lernprozessen verwendet wird, ist das so genannte »Power Law of Practice«. Es beschreibt, dass sich Kenntnisse zur Aufgabebearbeitung anfangs immer sehr schnell erweitern, während selbst kleine Verbesserungen später mit einem beträchtlichen Aufwand verbunden sind [Johnson/Bellman/Lohse 2002]. Die Methoden zur Aufgabebearbeitung werden am Anfang besonders stark verbessert, mit der Zeit werden Verbesserungen aber immer schwieriger. Diese Arbeit beschäftigt sich genau mit dieser Phase, in der Verbesserungen offenbar am schwierigsten zu erreichen sind.

2.4. »Intuitive Benutzbarkeit«

Sehr oft wird in der Öffentlichkeit eine »intuitive Benutzbarkeit« für Softwareanwendungen gefordert und häufig liest man in Computerzeitschriften von Beurteilungen, die Anwendungen eine »intuitive Bedienung« bescheinigen oder in Abrede stellen. Wenn nun eine Softwareanwendung intuitiv zu bedienen ist, wären damit nicht alle Probleme, den Umgang mit ihr zu erlernen, hinfällig? Ob Einarbeitung oder Weiterlernen im täglichen Umgang, muss man sich nicht einfach nur darum kümmern, die Software intuitiv bedienbar zu gestalten?

In diesem Abschnitt wird kurz dargelegt, dass diese Fragen gar nicht gestellt werden können. Dazu werden die Aussagen mehrerer Autoren zusammengefasst, die erklären, dass man von einer »intuitiven Benutzbarkeit« gar nicht sprechen kann. Der Begriff erzeugt nämlich den Eindruck, dass allein menschliche Intuition ausreicht, um eine Softwareanwendung zu bedienen, und dass eine Einarbeitung oder eine Art von Mitdenken überflüssig sind. Etwas wird als »intuitiv« bezeichnet, wenn man es ohne jegliche Vorkenntnisse und ohne kognitiven Aufwand sofort versteht [Raskin 94]. Z.B. wird der Gebrauch einer Computermaus oft als »intuitiv« bezeichnet. Doch selbst bei einem so einfachen Eingabegerät passt dieser Begriff nicht. Die Verwendung einer Maus ist für neue Nutzer, die noch keine Computermaus gesehen oder verwendet haben, zuerst sehr problematisch. Selbst wenn die Funktionsweise nach einer sehr kurzen Erklärung oder auch durch bloßes Zusehen korrekt erkannt wird, kann die Bedienung in der Anfangszeit sehr schwierig sein [Nielsen 93a].

Wenn jemand die Bedienung einer Anwendung oder eines Gerätes als »intuitiv« bezeichnet, ist viel eher »bekannt« gemeint (im englischen Originalartikel heißt es »familiar«). »Bekannt« soll die Verwendung von bereits bestehendem, gelerntem Vorwissen beschreiben. Eine Computermaus wird deswegen als »intuitiv benutzbar« bezeichnet, weil dieses Eingabegerät inzwischen so weit verbreitet ist und das Bedienkonzept so leicht verständlich ist, dass ihr Gebrauch mittlerweile eben von so vielen Menschen gelernt wurde. Treffender wäre auch hier die Bezeichnung »bekannt«. Es besteht das Missverständnis, gelernte Gewohnheiten in einem neuen Kontext, bei einem neuen Programm »intuitive Bedienung« zu nennen. Der ursprüngliche Lernaufwand wird außer Acht gelassen. Auch einfache Interaktionstechniken wie Doppelklicken und Drag-and-Drop sind nicht intuitiv, sondern müssen erst erlernt werden. Sie sind nur sehr weit verbreitet und bei Computernutzern allgemein bekannt [Tognazzini 98b].

Ironischerweise behindert das allgemeine Verständnis von »intuitiv« sogar die Entwicklung innovativer Anwendungen. Wenn nämlich etwas neu ist, ist es unbekannt, ungewohnt und anders, weswegen es als »nicht-intuitiv« abgelehnt wird, obwohl es vielleicht sogar viel schneller erlernbar ist als das Altbekannte. So wird die Verbreitung von Neuerungen erschwert, auch wenn sie dem als »intuitiv« bezeichneten Bekanntem überlegen sind [Raskin 2000].

2.5. Allgemeine Anforderungen an die Gebrauchstauglichkeit

In diesem Abschnitt werden die allgemeinen Anforderungen an die Gebrauchstauglichkeit von Softwareanwendungen zusammengefasst, die für das Weiterlernen bedeutsam sind.

Vielfach wird auf die Probleme von mehreren Modi in einer Benutzungsschnittstelle hingewiesen (z.B. [Graesser/Murray 90], [Sebrechts/Marsh/Furstenburg 90], [Nielsen 93a], [Selber/Johnson-Eilola/Mehlenbacher 97]). Eine Anwendung ist in einem bestimmten Modus, wenn die Auswirkungen einer Geste gleich sind. Wenn dieselbe Geste andere Auswirkungen hat, befindet sich die Anwendung in einem anderen Modus. Modi erzeugen Probleme, weil sie dazu führen, dass gewohnte Aktionen unerwartete Auswirkungen haben. Ein Beispiel für so ein Problem, das durch einen Modus erzeugt wird, lässt sich leicht bei der weit verbreiteten Tabellenkalkulation Microsoft Excel nachvollziehen. Eine Datei lässt sich über das entsprechende Icon nicht schließen, wenn sich die Anwendung im Editier-Modus befindet.

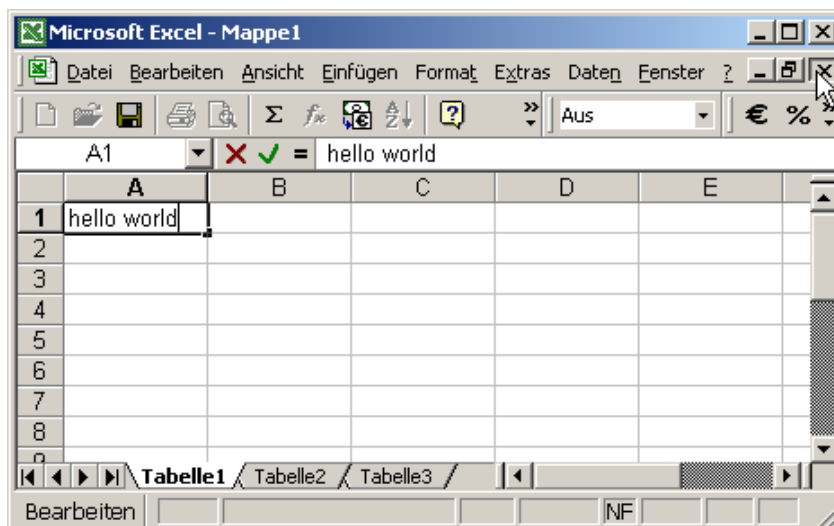


Abb. 2–2: Microsoft Excel im Editier-Modus

In Abb. 2–2 ist dargestellt, dass die Anwendung Tastatureingaben erwartet, um die Tabelle mit Inhalt zu füllen. In diesem Modus ist ein Mausklick auf das Icon zum Schließen der Datei (im Fenster oben rechts) ohne Wirkung. Erst wenn der Editier-Modus verlassen wurde, schließt ein Klick auf das Icon die Datei wie gewohnt (vgl. Abb. 2–3). Dieselbe Geste hat also unterschiedliche Auswirkungen, abhängig davon, ob die Anwendung im Editier-Modus ist oder nicht.

Systemrückmeldungen und Anzeigen von Systemzuständen reichen nicht aus, um Modusfehler zu eliminieren, selbst wenn die Anzeigen sich an dem Ort befinden, auf den sich die Aufmerksamkeit des Nutzers richtet. Oftmals werden Modi durch verschiedene Cursorformen angezeigt, doch die Aufmerksamkeit eines Nutzers richtet sich auf das selektierte Objekt und nicht auf die Form des Cursors. Wenn die Aufmerksamkeit des Nutzers nicht auf den aktuellen Systemzustand gerichtet ist und die Anwendungen Modi enthält, wird der Nutzer manchmal Fehler machen, weil seine Aufmerksamkeit eben nicht der aktuelle Modus ist. Auch sehr deutliche Modus-Indikatoren sind keine Lösung, denn so wird die Aufmerksamkeit weg von der eigentlichen Aufgabe und hin zum Systemzustand gelenkt. Eine Anwendung sollte ohne Modi auskommen, damit die Nutzer so weit wie

möglich die Kontrolle über das System behalten. Das gilt natürlich ebenso für Formen der Lernunterstützung [Raskin 2000]. Besonders beim Erlernen erzeugen Modi das Problem, dass sie den Lernprozess mit einem notwendigen Lernschritt blockieren. Der Nutzer kann bei einem Modus die Lernreihenfolge nicht mehr frei festlegen, sondern ihm wird ein Lernschritt zwingend vorgegeben. Das birgt besondere Frustrationsgefahr [Lassig 2001].

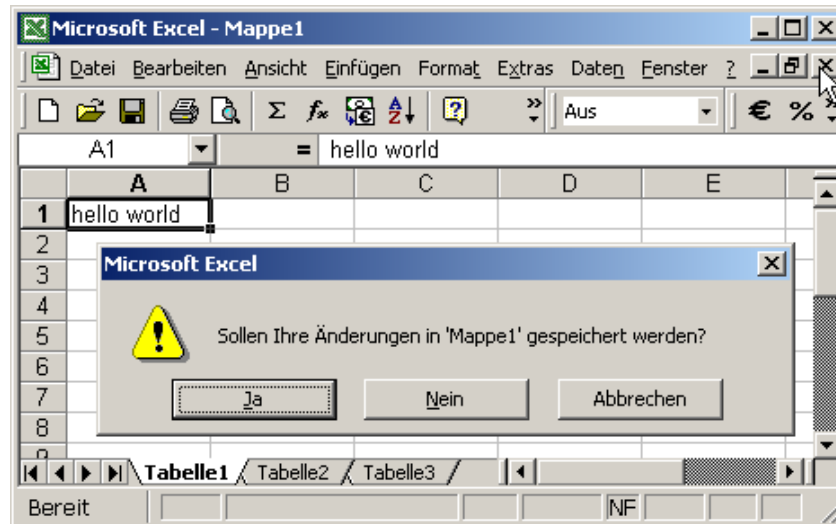


Abb. 2—3: Schließen einer Datei in Microsoft Excel

Zu den Inhalten von Lernhinweisen wird gefordert, dass sie sowohl deklaratives als auch prozedurales Wissen vermitteln, wobei sich die Erklärungen an den typischen Aufgaben der Nutzer orientieren sollten und nicht an den Systemfunktionen (siehe z.B. [Graesser/Murray 90], [Pierotti 98], [Tomasi/Mehlenbacher 99], [Houchin 2001]). Deklaratives Wissen wird durch Beschreibungen von Sachverhalten, Gegenständen, Zuständen, Prozessen und Zusammenhängen vermittelt. Der Nutzer erfährt, wofür die verschiedenen Anwendungselemente dienen. Mit prozeduralem Wissen erfahren die Nutzer, wie sie bestimmte Aufgaben tatsächlich ausführen. Sie werden angeleitet, etwas zu tun oder auch zu unterlassen [Donker 2002].

Auch die Formulierungen von Texten spielen eine wichtige Rolle. So stand am Beginn der Entwicklung des Microsoft Office-Assistenten (siehe S. 29ff) die Erkenntnis, dass die Nutzer Schwierigkeiten bei dem effektiven Auffinden von Unterstützung hatten. Die Probleme der Nutzer beim Zugriff auf die Lösungen oder dem Auffinden von Funktionalitäten rührten eindeutig von Problemen mit der Terminologie her [Heckerman/Horvitz 98]. Zur Minimierung solcher Verständnisprobleme sollte die Sprache aus dem Anwendungsfeld der Nutzer verwendet werden. Damit Nutzer Erklärungen gut verstehen, ist eine gegliederte Darstellung und die Verwendung einfacher Formulierungen mit kurzen Sätzen wichtig. Je länger ein Satz ist, desto mehr Informationen enthält er, desto komplexer wird der Aufbau sein und desto schwerer wird er daher verständlich sein [Donker 2002]. Die richtige Sprache, um Exploration anzuregen, ist etwa »Versuchen Sie es ruhig.«, »Probieren Sie es und sehen sie selbst die Ergebnisse.« oder »Beobachten Sie, was passiert.« Formulierungen wie »Sie können auch...« und »Eine andere Möglichkeit ist...« sollten vermieden werden [Van der Meij/Carroll 98].

Einleitungen, Überblicke und Zusammenfassungen sollten auf das absolut Nötigste reduziert werden. Erfahrene Nutzer überspringen diese Passagen oft, während Neulinge sich darin bloß festlesen. Wenn ein Überblick technische Begriffe oder Kommentare zu Optionen enthält, wird er für den Neuling zu bedeutungslosem technischen Gerede. Um Erklärungen so kurz wie möglich zu halten, sollten sie nichts Historisches oder Philosophisches enthalten, da sich Nutzer nicht für die Entwicklungsgeschichte eines Produktes oder Designentscheidungen interessieren. Ebenso kann man auf technische Details verzichten, da Nutzer sich nicht für interne Mechanismen einer Anwendung interessieren [Lewis/Rieman 93].

In Übereinstimmung mit den Folgerungen aus den lernpsychologischen Grundlagen fordern zahlreiche Autoren, Softwareanwendungen explorierbar zu gestalten. Sehr umfassend beschäftigt sich [Paul 93]

mit diesem Thema. Die häufigste Empfehlung ist, den Nutzern Undo- und Redo-Funktionalität zur Verfügung zu stellen (vgl. [Brodbeck/Rupietta 94], [Pierotti 98], [Lassig 2001]).

2.5.1. Minimalismus

Es wurde bereits deutlich, dass Konstruktivismus und aktives Lernen wichtige Grundlagen für die Erleichterung kontinuierlichen Weiterlernens bilden. Diese zwei Schlüsselprinzipien der kognitiven Psychologie werden im minimalistischen Nutzermodell miteinander verbunden [Redish 98]. Wesentliche Kennzeichen des Minimalismus sind Aufgabenorientierung und Kürze. Richtlinien für die Gestaltung von Softwareanwendungen nach diesem Modell finden sich vor allem in [Carroll/van der Meij 98] und [Van der Meij/Carroll 98]:

- sofortige Möglichkeit anbieten, aktiv zu handeln
Minimalismus zielt darauf ab, dass die Nutzer schnell mit realen Aufgaben beginnen. Am Anfang sollten daher gleich Handlungsanweisungen und grundlegende prozedurale Informationen in einer sehr knappen Form stehen. Erklärungen sollten nicht mit Konzepten zur Funktionsweise der Anwendung oder Semantiken aus der Anwendungsdomäne beginnen. Konzeptuelle Informationen sind nur dort enthalten, wo es für ein adäquates Verständnis und die Ausführung von Aufgaben notwendig ist. Man sollte den Nutzern zu Beginn weniger zu lesen und mehr zu tun geben. Die Erfüllung sinnvoller Aufgaben ist die Basis für anschließende konzeptuelle Einzelheiten. Allgemein helfen Erklärungen mehr nach der Vervollständigung einer Tätigkeit, da der Nutzer sich dann bereits Vorwissen angeeignet hat, das beim Verstehen der Erklärungen hilft.
- Exploration und Ideenfindung anregen und unterstützen
Wenn die notwendige Zeit zum Lesen minimiert wird, sollte stattdessen Exploration angeregt werden [Hackos 98]. Das Vorbereiten von und Hinweisen auf Exploration bildet für die Nutzer eine starke Anregung, sich auch tatsächlich explorativ mit der Anwendung auseinanderzusetzen. Dabei muss man jedoch den richtigen Mittelweg zwischen ungeleiteter Exploration und zu detaillierten Übungsschritten finden.
- Steuerung durch den Nutzer gewährleisten
Entwickler können Formen der Unterstützung gestalten, sollten sie den Nutzern aber nicht aufdrängen. Unterstützungsfunktionen sollten erst dann erscheinen, wenn sie vom Nutzer angeschaltet werden.
- Lehraktivitäten auswählen oder entwerfen, die die Struktur realer Aufgaben widerspiegeln
Durch das Bereitstellen von Vorlagen können die Erledigung realer Aufgaben ermöglicht und umfangreiche Vorarbeiten der Nutzer reduziert werden.
Ein minimalistisches Handbuch sollte auf realen Aufgaben beruhen [Nielsen 93a]. Die Überschriften der Anleitungen sollten einen Überblick über die Hauptaufgaben einer Domäne geben.
- Fehlererkennung und -behebung unterstützen
Nutzerfehler werden im Minimalismus wie auch beim explorativen Lernen gleichermaßen sehr stark berücksichtigt. Beide Lernformen behandeln Fehler eher als natürliche Ereignisse, aus denen Nutzen gezogen werden sollte. Dementsprechend richtet sich ein wesentlicher Teil der Aufmerksamkeit auf die Unterstützung der Fehlererkennung und -behebung. Die Auseinandersetzung mit Fehlern steigert das Verständnis der Nutzer [Farkas 98]. Fehlerinformationen sollten angeboten werden, wenn Aktionen fehleranfällig sind – z.B. wenn der Nutzer keine oder zu wenig Kenntnisse hat oder wenn eine Aktion von Standardabläufen abweicht – oder Korrekturen nur schwer vorzunehmen sind, weil beispielsweise die Ergebnisse einer Aktion nur schwer interpretierbar sind. Sie sollten die Erkennung, Diagnose und Korrektur ermöglichen und immer direkt an dem Ort platziert werden, an dem ein Fehler auftreten kann. Dies erleichtert auch das Explorieren der Fehlerbehebung.
- sich kurz fassen und nicht alles erklären
Man sollte kleine, übersichtliche Einheiten anbieten, deren Umfang nicht abschrecken darf, sondern zusätzlich motivieren soll. Information, die leicht selbst zu erschließen ist, sollte weggelassen werden. Durch die Kürze vermitteln Anleitungen den Nutzern auf den ersten

Blick, dass ihre Aufgaben keinen gewaltigen Aufwand erfordern. Da nicht alles erklärt wird, regen die Anleitungen die Nutzer an, die relevanten Vorkenntnisse zu aktivieren und sich mehr auf das eigene Denken zu verlassen.

Wenn Informationseinheiten auf nur wenige, kurze Absätze beschränkt werden und jede Einheit deutlich betitelt wird, können die Nutzer selbst entscheiden, wie viel Zeit sie mit dem Lesen von Anleitungen verbringen möchten, statt mit der Anwendung zu experimentieren [Hackos 98].

Mit all diesen Richtlinien orientieren sich minimalistische Anleitungen sehr stark an den Lernbedürfnissen von Nutzern. Darüber hinaus berücksichtigt der Minimalismus das Produktivitätsparadox. Das Produktivitätsparadox ist ein Prinzip menschlicher Kognition. Menschen möchten sinnvolle Ziele erreichen, Dinge ausprobieren und immer produktiv sein. Um aber langfristig produktiv zu sein, sind manchmal Aktivitäten nötig, die kurzfristig nicht oder kaum produktiv sind – z.B. Lesen von Dokumentation. Lesen von Dokumentation ist immer eine Ablenkung von der gerade produktiven Arbeit. Man muss also das Bedürfnis der Nutzer, aktiv zu handeln, ins Gleichgewicht mit dem Bedürfnis nach Wissensvermittlung bringen (vgl. [Carroll/van der Meij 98] und [Van der Meij/Carroll 98]). Genau dies soll eine Unterstützung von aktivem, explorativem Lernen wie im Minimalismus propagiert leisten. Der Ansatz mit einer Anleitung wie etwa »Lesen Sie alles genau durch, bevor sie etwas tun!« ist zwar auch ein Versuch, das Problem zu lösen, aber ein eher verzweifelter [Rosson/Carroll 2002].

2.5.2. Adaptivität und Adaptierbarkeit

Bei den Betrachtungen zur Anpassung von Systemen wird unterschieden, wer Systemanpassungen veranlasst. Wenn das System Anpassungen selbst, automatisch veranlasst, ist es adaptiv. Eine abgeschwächte Form adaptiver Systeme ist die, in der das System dem Nutzer nur einen Vorschlag für eine Anpassung gibt, die der Nutzer aber noch bestätigen muss. So behält er wenigstens die Kontrolle. Adaptive Systeme beobachten im Allgemeinen die Nutzungshistorie und erstellen ein Nutzermodell, um geeignete Anpassungen zu identifizieren. Ein wichtiger Aspekt ist dabei, dass aus Datenschutzgründen der Nutzer (und auch niemand sonst) das vom System über ihn angelegte Modell einsehen und auf Wunsch löschen können muss. Ein viel einfacherer Fall von adaptivem Verhalten ist es, wenn das System beim Start den Zustand wieder herstellt, in dem es sich befunden hat, als es das letzte Mal beendet wurde – und z.B. die zuletzt geöffneten Dateien wieder anzeigt. In diesem Fall ist das Anlegen eines Nutzermodells nicht nötig, so dass diese Form der Adaption allgemein als unkritisch angesehen wird.

Systeme, bei denen der Nutzer die Anpassungen aktiv von sich aus veranlasst, werden adaptierbar genannt. Eine häufige Form der Adaptierbarkeit sind die typischen Konfigurationsmöglichkeiten für den Nutzer, persönliche Präferenzen zur Benutzungsoberfläche einzustellen. So kann man oft die Platzierung von Bedienelementen oder die Ausführlichkeit von Systemausgaben anpassen (siehe z.B. [Preim 99]). Zu adaptierbaren Systemen werden aber auch tief greifende Änderungen an der Systemfunktionalität gezählt [Paul 93].

Die Meinungen zu diesem Thema gehen zwischen Befürwortern und Gegnern von Anpassbarkeit weit auseinander, wobei zwischen adaptiven und adaptierbaren Systemen wieder unterschieden werden muss.

Probleme werden vor allem bei adaptiven Systemen gesehen. Automatische Anpassungen durch das System bevormunden den Nutzer, erschweren die Exploration, führen zu Orientierungsproblemen, die Inkonsistenz widerspricht einer geforderten Erwartungskonformität und das aufgestellte Nutzermodell kann theoretisch missbraucht werden (vgl. [McGrenere/Moore 2000] und [Paul 93]). Wenn eine Anpassung nicht den Bedürfnissen des Nutzers entspricht, wird der Nutzer sich gestört fühlen und weniger effizient mit dem System arbeiten [Encarnaçao 97].

Bei adaptierbaren Systemen wird als Vorteil angeführt, dass besonders die Reduzierbarkeit von Funktionalität das explorierende Kennen lernen eines Systems unterstützt. Adaptierbare Systeme reduzieren Fehlersituationen, sie sind transparenter und steuerbarer. Auf diese Weise wird Komplexität durch die Nutzer kontrollierbar, wodurch sich die Lernförderlichkeit erhöht [Paul 93].

[Raskin 2000] aber spricht sich allgemein nicht nur gegen adaptive sondern auch gegen adaptierbare Systeme aus. Die Nutzer sollen von einem System ausschließlich bei der Erledigung ihrer Aufgaben unterstützt werden und nicht zu einer Auseinandersetzung mit dem System selbst verleitet werden. Die Anwendung sollte so wenig komplex wie möglich sein, wozu auch der Verzicht auf zu viele, zu komplizierte zusätzliche »Lernhilfen« zählt. Selbst einfache persönliche Einstellungen sollten den Nutzern nicht möglich sein. Sonst stellten sich eine ganze Reihe von negativen Effekten ein:

- Eine Personalisierung durch Nutzer liefert oft Ergebnisse, die nicht optimal sind, wodurch die Effizienz sinkt.
- Die Dokumentation passt eventuell nicht mehr zur modifizierten Benutzungsschnittstelle.
- Personalisierungsoptionen erhöhen die Komplexität einer Anwendung.
- Sie laden zur Zeitverschwendung ein, wenn man von der Bearbeitung der Aufgaben abgelenkt wird und stattdessen mit den Einstellungen »rumspielt«.
- Sie gefährden die Stabilität eines Systems.
- Sie untergraben die Ausprägung von Gewohnheiten.
- Sie führen zu mehr Problemen beim Support per Telefon oder E-Mail.

In diesem Zusammenhang lehnt [Raskin 2000] auch das Anbieten unterschiedlicher Benutzungsschnittstellen für Neulinge und Experten ab, worauf im folgenden Abschnitt näher eingegangen wird. Das Prinzip dieser Haltung ist, dass sich Benutzungsschnittstellen mit der Zeit überhaupt nicht ändern sollten.

2.5.3. Unterscheidung von Neulingen und Experten

Nutzer unterscheiden sich und haben viele verschiedene Bedürfnisse. Die drei Dimensionen, in denen sich die Erfahrungen der Nutzer unterscheiden, sind: Allgemeine Computerkenntnisse, Erfahrungen mit der Verwendung eines speziellen Systems und Verständnis der Aufgabendomäne [Nielsen 93a]. Bei der traditionellen Unterscheidung von Neulingen und Experten bleibt jedoch unklar, nach welchen Kriterien die Kenntnisse der beiden Nutzerkategorien beurteilt werden. Die Begriffe können daher irreführend sein und sind zumindest interpretationsbedürftig [Selber/Johnson-Eilola/Mehlenbacher 97].

Nun fordert [Nielsen 93a] zwar, dass Neulinge und Experten z.B. durch vielfältige Interaktionsstile wie etwa Tastaturkürzel unterschiedlich unterstützt werden, gibt gleichzeitig aber zu bedenken, dass sich durch die simultane Ausrichtung auf mehrere Zielgruppen die Komplexität einer Anwendung erhöht. Und selbst ein so genannter »Experte« kann bezüglich vieler Teile des Systems, die er normalerweise nicht verwendet, ein »Neuling« sein.

Auch wenn sich Nutzer unterscheiden und viele verschiedene Bedürfnisse haben, so gibt es doch viele Merkmale, die allen Nutzern gemeinsam sind. Bevor man eine Anwendung an die Unterschiede zwischen den Nutzern anpasst, kann man bei der Gestaltung einer Anwendung Nutzen daraus ziehen, was den Menschen bezüglich Anforderungen an das Anwendungsdesign gemeinsam ist [Raskin 2000].

2.6. Rechtliche Vorgaben und Standards

An dieser Stelle wird auf die verschiedenen Normen und Gesetze zur Software-Ergonomie und Gebrauchstauglichkeit eingegangen. Das Thema »Erlernen von Softwareanwendungen« wird in diesen Dokumenten unter dem Begriff »Lernförderlichkeit« beschrieben. In der internationalen Norm ISO 9241-10 zählt Lernförderlichkeit zu den sieben Gestaltungs- und Bewertungsgrundsätzen eines Dialogs – neben Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Steuerbarkeit, Erwartungskonformität, Fehlertoleranz und Individualisierbarkeit. Dort steht: »Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen des Dialogsystems unterstützt und anleitet.« [ISO 9241-10, S. 9] An gleicher Stelle werden mehrere Empfehlungen gegeben, wie so eine Unterstützung und Anleitung aussehen kann. U.a. wird empfohlen, relevante Lernstrategien wie z.B. verständnisorientiertes Lernen, »Learning-by-doing« und Lernen am Beispiel zu unterstützen. Diese Empfehlungen werden wiederum

durch Beispiele präzisiert. So sollen Nutzer ermutigt werden, »zu experimentieren, in unterschiedlichen Situationen Beispiele durchzuspielen, ›Was wäre wenn?‹-Alternativen anzuwenden«, um »Learning-by-doing« zu unterstützen [ebd.].

Da sie in der Norm ISO 9241-10 festgeschrieben sind, werden diese Aspekte auch in entsprechenden Leitfäden berücksichtigt, mit denen die Einhaltung der software-ergonomischen Grundsätze geprüft werden soll. So fragen der 100-Fragen-Test nach Riehenhagen und der Benutzungsfragebogen ISONORM 9241/10 zur Lernförderlichkeit u.a., ob die Software dazu ermutigt, auch einmal Neues auszuprobieren (beide siehe [TBS 97]). Ebenso beschreibt der ErgoNorm-Benutzerfragebogen zu »Arbeit & Software« eine Anwendung als lernförderlich, wenn sie es ermöglicht, »selbständig einfach mal ›rumzuprobieren‹, ohne dass [die Nutzer] Angst haben müssen, etwas ›kaputt‹ zu machen« [DATech 2001, S. 106]. Dieser Fragebogen analysiert den Aspekt des gefahrlosen Ausprobierens weiter, indem nach eventuellen »Strafen« gefragt wird, die Nutzer nach »Rumprobieren« von der Anwendung bekommen haben.

In der Unfall-Verhütungsvorschrift der Verwaltungs- und Berufsgenossenschaft zu »Arbeit an Bildschirmgeräten« (kurz: UVV VBG 104) von 1995 gibt es einen eigenen Paragraphen zur Lernförderlichkeit: »Die Software muß so gestaltet sein, daß sie die Benutzer während des Lernprozesses in erforderlichem Umfang unterstützt und anleitet.« [UVV_VBG_104 95, § 36] Dieser Paragraph wird durch eine zugehörige Durchführungsanweisung näher erläutert, in der »Lernstrategien wie verständnisorientiertes Lernen, Lernen durch Handeln, Lernen am Beispiel« als mögliche Unterstützungsformen zur Selbständigkeit der Nutzer aufgezählt werden [ebd.].

Die EU-Bildschirmrichtlinie 90/270/EWG von 1990 und deren Umsetzung in nationales Recht in Form der staatlichen »Verordnung über Sicherheit und Gesundheitsschutz bei der Arbeit an Bildschirmgeräten« von 1996 behandeln das Thema Lernförderlichkeit nicht ausdrücklich. Allenfalls wird in beiden Texten die Anwendung von Grundsätzen der Ergonomie gefordert (siehe dazu [EU 90] und [BildscharbV 96]).

Aspekte, die für das Erlernen von Softwareanwendungen eine Rolle spielen, werden auch noch in der ISO-Norm 9241-13 über Benutzerführung angesprochen. Diese Norm bezieht sich aber nur auf Interaktions-Komponenten, die Nutzer »bei der Behebung von Fehlersituationen unterstützen« [ISO 9241-13, S. 4]. Es werden Anforderungen an Eingabeaufforderungen, Rückmeldungen, Statusinformationen, Fehlermanagement und Online-Hilfe beschrieben. Die Norm bezieht sich explizit nicht auf andere Arten der Anleitung für Nutzer wie Online-Lernkurse, Online-Dokumentation und intelligente Systemunterstützung. Ihr Fokus bezüglich von Lernaspekten ist daher enger als der dieser Arbeit.

Zusammenfassend kann gesagt werden, dass auch in den einschlägigen Standards und rechtlichen Vorgaben zur Softwareergonomie gefordert wird, dass Anwendungen explorationsfreundlich gestaltet werden.

2.7. Folgerungen für das Anwendungsdesign zur Erleichterung kontinuierlichen Weiterlernens

Die in diesem Kapitel gesammelten Erkenntnisse werden nun zusammengefasst, um Anforderungen an das Anwendungsdesign aufzustellen, die kontinuierliches Weiterlernen erleichtern. Diese Empfehlungen eignen sich für Anwendungen, die interaktiv und nutzergeführt sind.

- Die Unterstützung für die Nutzer durch das Anwendungssystem sollte sich an den Prinzipien des Minimalismus orientieren.
- Das System sollte Möglichkeiten zum aktiven Lernen bieten und unterschiedliches Vorwissen der verschiedenen Nutzer berücksichtigen.
- Die Hinweise müssen sich an den Aufgaben der Nutzer statt an Systemfunktionen orientieren und kurz gefasst sein.

- Den Nutzern sollten vielfältige Gelegenheiten angeboten werden, explorativ mit dem System umzugehen. Zu solch einer Exploration sollten die Nutzer durch entsprechende Anregungen auch ermutigt werden.
- Damit die Nutzer die angebotenen Lernmöglichkeiten auch annehmen, muss man sich um eine Förderung der intrinsischen Motivation bemühen.
- Um die Nutzer nicht unnötig zu verwirren, sollte die Benutzungsschnittstelle eines Systems mit möglichst wenig Modi auskommen.
- Zur Bereitstellung adaptierbarer Benutzungsschnittstellen können keine eindeutigen Empfehlungen gegeben werden, da die Meinungen hierzu auseinander gehen. Übereinstimmung besteht jedoch darin, dass ein System keinesfalls auf eine Weise adaptiv sein sollte, die die Verwendung eines Nutzermodells erfordert.

3. Vorhandene Lösungen zur Unterstützung von Anwendern

»Dieser unbeliebte Helfer hat die irritierende Angewohnheit, immer zu den falschesten Zeiten aufzutauchen und ungefragt seine Hilfe anzubieten, nur um einen dann mit der vergeblichen Suche danach, wie man ihn wieder loswerden kann, vollends in den Wahnsinn zu treiben.« [Rupp 2001]
Katja Rupp über den Microsoft Office-Assistenten

Zur Analyse des aktuellen Standes, wie Systeme schon jetzt kontinuierliches Weiterlernen erleichtern, stelle ich zunächst eine neue Klassifikation von Lernkomponenten, Hilfe- und Assistenzsystemen vor. Diese bildet den Ausgangspunkt für die anschließende Besprechung der wichtigsten bisher bestehenden Lösungen.

3.1. Klassifikation anwenderunterstützender Systeme

Hilfesysteme unterstützen Nutzer in zwei Bereichen. Beim *Wissenserwerb* erlernen Nutzer neue Sachverhalte und beseitigen Wissensprobleme. Beim *Fehlermanagement* informieren sich Nutzer über Informationen und Funktionen zur Behebung von Fehlern [Brodbeck/Rupietta 94]. In dieser Arbeit liegt der Schwerpunkt auf den Aspekten zum Wissenserwerb. Es bietet sich daher an, zunächst bestehende Klassifikationen von Hilfesystemen zu betrachten.

3.1.1. Gebräuchliche Modelle

Allgemeine Ziele bei der Aufstellung einer Klassifikation sind die Förderung der Diskussion über ein Thema und die Erleichterung von Weiterentwicklungen [Raskin 2000]. Unter diesem Blickwinkel erweisen sich gebräuchliche Klassifikationen von Hilfesystemen oft als wenig hilfreich. Im Allgemeinen werden folgende vier Merkmale mit jeweils zwei Ausprägungen unterschieden (Bezeichnungen nach [Brodbeck/Rupietta 94]):

- Initiative: passiv oder aktiv
Eine passive Komponente muss erst vom Nutzer aufgerufen werden, eine aktive Komponente wird von der Anwendung selbständig aufgerufen.

- Adressatenbezug: benutzerunabhängig oder benutzerabhängig
Verwendung eines Nutzermodells, bezieht sich sowohl auf individuelle Nutzer als auch im größeren Sinne auf Nutzergruppen.
- Kontextbezug: kontextunabhängig oder kontextabhängig
Unterscheidung, ob der aktuelle Fokus des Nutzers oder der aktuelle Systemstatus berücksichtigt wird, oder nicht.
- Inhalt: funktionsorientiert oder aufgabenorientiert
Unterscheidung zwischen Ansätzen, die sich an den Funktionen des Systems oder den typischen Aufgaben in der Anwendungsdomäne orientieren.

Unglücklicherweise haben sich für die Ausprägungen auch andere Bezeichnungen eingebürgert. Die Initiative wird auch als »benutzerinitiiert« oder »systeminitiiert«, der Adressatenbezug ebenso als »uniform« oder »individuell«, und der Kontextbezug wiederum als »statisch« oder »dynamisch« bezeichnet (z.B. in [Göbel 2001] und [Heinecke 98]). Gegenüber einem einheitlichen Sprachgebrauch erschweren solche Synonyme eine Diskussion.

Man kann zwar alle bestehenden Hilfesysteme wie auch Lernsysteme und Assistenzkomponenten einfach in dieses Schema einordnen, erfährt dabei aber nur wenig über das jeweilige System. Beispielsweise sind sowohl Tutorien als auch Über-Sichten und Explorationskarten passiv, benutzerunabhängig, kontextunabhängig und aufgabenorientiert. Dieses Klassifikationsschema kann die Unterschiede dieser drei Komponenten nicht vermitteln.

Auch zur Anregung von Weiterentwicklungen ist ein Schema dieser Art wenig geeignet. Es gibt wenig Bezug zu konkreteren, charakteristischen Implementierungsdetails und Designvorschläge sind selbst auf einer abstrakten Ebene nicht ableitbar. Diese Schwächen sollen mit der neuen Klassifikation behoben werden.

Neben reinen Hilfesystemen, in deren Mittelpunkt die direkte Unterstützung des Nutzers im aktuellen Anwendungskontext steht, werden hier auch andere Unterstützungsformen wie Tutorielle Systeme betrachtet, bei denen das Lernen selbst am wichtigsten ist und nicht unbedingt an einen Arbeitskontext gebunden sein muss (vgl. [Brodbeck/Rupietta 94]).

Assistenzsysteme nach [Wandke 2002] haben eine ähnliche Ausrichtung wie Hilfesysteme. Sie dienen dazu, prinzipiell vorhandene Funktionen, die den Nutzern jedoch unbekannt oder unzugänglich sind, verfügbar zu machen. Dies beschreibt genau die Motivation, kontinuierliches Weiterlernen zu erleichtern. Assistenzsysteme werden nach vier Dimensionen klassifiziert:

- Unterstützungsgrad: Verschiedene Stufen von Automatisierung, von informativen bis zu vollautomatischen Komponenten
- Handlungsphasen: Von der Motiv- und Zielbildung, über die Informationsaufnahme, -analyse, Auswahl einer Aktion und deren Ausführung bis zur Kontrolle der Auswirkungen
- Initiative: passiv oder aktiv
- Anpassbarkeit: nicht anpassbar, adaptierbar oder adaptiv

Der Fokus von Assistenzsystemen ist aber wie bei Hilfesystemen eng an die aktuelle Aufgabe und den jeweiligen Anwendungskontext gebunden.

3.1.2. Merkmale für eine neue Klassifikation

Die hier vorgestellte neue Klassifikation soll einen hohen praktischen Nutzen für die Entwicklung von Systemen zur Erleichterung kontinuierlichen Weiterlernens bieten. Bestehende Ansätze werden dort genauer und konkreter beschrieben. Durch den relativ feinen Detaillierungsgrad eignet sich diese Klassifikation gut für die Generierung neuer Ansätze. In dieser Klassifikation werden die Merkmale

- Fragegenerierung bzw. Problembeschreibung,
- Zuordnung der Antwort,
- Art der Antwort,
- Kommunikation und
- Ausgabe

unterschieden. Diese Begriffe werden im Folgenden erklärt und zu jedem Merkmal werden die verschiedenen Ausprägungen aufgelistet.

Fragegenerierung, Problembeschreibung

Wie wird die Frage formuliert, die beantwortet werden soll? Wie wird das Problem beschrieben, das gelöst werden soll? Wer macht das?

- [1] keine: Es gibt keine spezielle Frage, die die Anwendung beantworten soll.
- [2] Nutzer formuliert Frage: Der Nutzer hat eine spezielle Frage, die er stichwortartig oder ausformuliert an das System richtet.
- [3] Nutzer trifft Auswahl: Die Anwendung bietet dem Nutzer eine beschränkte Anzahl von vorbereiteten Erklärungen bzw. Hinweisen an, aus denen der Nutzer die (hoffentlich) passende auswählen kann.
- [4] Nutzer zeigt auf Interaktionselement: Der Nutzer deutet auf ein spezielles Element der Benutzungsschnittstelle, was von der Anwendung als Frage über dieses Element interpretiert wird.
- [5] System ermittelt aktuellen Kontext: Die Anwendung beobachtet ihren jeweils aktuellen Zustand und den Fokus des Nutzers. So sollen Situationen erkannt werden, die für den Nutzer problematisch sind, um dann eine passende Unterstützung zu geben.
- [6] System analysiert Nutzungshistorie: Die Anwendung protokolliert die gesamte, individuelle Nutzungshistorie eines Nutzers, analysiert diese und schließt daraus auf für den Nutzer nützliche Hilfestellungen.

Zuordnung Antwort

Wie wird die Antwort, der Hinweis, die Erklärung zu der gestellten Frage oder dem Problem gefunden?

- [1] per Zufall: Es wird das Zufallsprinzip angewendet.
- [2] Verzeichnis, Verschlagwortung, Volltextindizierung: Der Nutzer sucht die Antwort aus einer Reihe vorbereiteter Erklärungen über ein Inhaltsverzeichnis aus (gegebenenfalls mit Hilfe eines Schlagwortverzeichnisses oder einer Volltextsuche).
- [3] aktueller Fokus: Die Anwendung richtet sich bei ihrer Reaktion nach dem aktuellen Fokus des Nutzers.
- [4] nach Analyse: Es findet eine mehr oder weniger komplexe Analyse der Nutzungshistorie statt. Das Spektrum der möglichen Analysearten ist sehr breit. Es reicht von einfacher Analyse (z.B. Aufzeichnung von Befehlshäufigkeiten, siehe im Abschnitt 3.6 über Nutzer- und Expertenmodellierung, S. 32f) bis zu komplexen Verfahren (z.B. Nutzermodellierung in Bayes-Netzen mit Markov-Ketten wie beim Microsoft Office-Assistenten, siehe im Abschnitt 3.5 über Intelligente Hilfe, S. 29ff) oder sogar selbstlernenden Systemen.

Art der Antwort

Welchen funktionalen Bezug hat die Antwort zu der eigentlichen Anwendung? Die drei hier beschriebenen Stufen werden auch als Aufgaben von Assistenzsystemen beschrieben, die »informieren, zusätzliche Handlungsvorschläge unterbreiten oder auch selbständig erforderliche Handlungen ausführen« sollen [Bente/Krämer 2000, S. 1].

- [1] beschreibend: Dem Nutzer werden lediglich Erklärungen geliefert, bis auf die Ausgabe von textlichen Inhalten bleibt das System dabei passiv.
- [2] hinweisend: Es gibt Verweise zu Funktionen in der Anwendung, z. B. um dem Nutzer einen direkten Funktionsaufruf aus der Hilfefunktion heraus zu ermöglichen.
- [3] unterstützend: Das System geht dem Nutzer bei der Erledigung seiner Arbeit auf irgendeine Weise aktiv zur Hand, indem z. B. komplette Funktionen aus der Anwendung übernommen werden.

Kommunikation

Welche Form hat die Kommunikation zwischen dem Nutzer und dem Hilfesystem?

- [1] einseitig vom System: Das System gibt Meldungen aus, auf die der Nutzer nicht direkt antworten kann. Diese Form ist also ein Monolog.

- |2| doppelseitig: Doppelseitige Kommunikation zeichnet sich dadurch aus, dass sich jede Seite genau einmal äußert. Meist stellt der Nutzer eine Anfrage, auf die das System eine Antwort gibt. Dies kann natürlich mehrmals hintereinander erfolgen, aber die Antworten beziehen sich immer nur auf die direkt vorher getätigte Anfrage, so dass man Frage und darauf folgende Antwort jeweils isoliert betrachten kann. Eine lange doppelseitige Kommunikation kann darauf hinweisen, dass der Nutzer lange in dem System nach der Antwort sucht, die ihm endlich weiterhilft.
- |3| wechselseitig: In dieser Form treten Nutzer und System in einen längeren Dialog, in dem die Kommunikationspartner mehrmals hintereinander, abwechselnd antworten. Dabei beziehen sich die Fragen und vor allem alle Antworten auf den gesamten Kommunikationsverlauf, so dass man sie nicht losgelöst voneinander betrachten kann. Das System begleitet hier also den Nutzer über mehrere Dialogschritte bei der Ausführung seiner aktuellen Aufgabe.

Ausgabe

Wann und wo findet die Ausgabe statt?

- |1| unmittelbar, in Fokusnähe: Die Reaktion des Systems wird unverzüglich und auf den aktuellen Fokus gerichtet ausgegeben.
- |2| unmittelbar, in separatem Fenster: Die Reaktion oder Antwort erscheint sofort nach Anforderung von der Anwendung getrennt in einem eigenen, separaten Fenster (Fenster erster oder zweiter Ordnung).
- |3| verzögert, in separatem Fenster: Die Reaktion oder Antwort erscheint um eine gewisse Zeit verzögert und von der Anwendung getrennt in einem eigenen, separaten Fenster (Fenster erster oder zweiter Ordnung).
- |4| verzögert, außerhalb der Anwendung: Diese Ausgabeform hat eine große zeitliche und räumliche Distanz zur Anfrage.

3.1.3. Übersichtstabelle

Die folgende Tabelle bietet einen Überblick über Lernkomponenten, Hilfe- und Assistenzsysteme, die auch als Ansätze zur Erleichterung kontinuierlichen Weiterlernens bezeichnet werden können. Diese Ansätze werden in den folgenden Abschnitten beschrieben und bewertet.

In dieser Klassifikation erkennt man nun auch die Unterschiede zwischen Tutorien, Über-Sichten und Explorationskarten. Diese werden bei der Art der Antwort (Über-Sichten sind hinweisend, während die beiden anderen Komponenten beschreibend sind) und v.a. bei der Kommunikation deutlich, bei der alle drei Komponenten eine andere Ausprägung haben.

Das als Beispiel angegebene OWL-System von Frank Linton wird im entsprechenden Abschnitt über Nutzer- und Expertenmodellierung (S. 32f) vorgestellt.

Ansatz	Fragegenerierung, Problembeschreibung	Zuordnung Antwort	Art der Antwort	Kommunikation	Ausgabe	Beispiel
Online-Handbücher	Nutzer formuliert Frage	Verzeichnis, Verschlagwortung, Volltextindizierung	beschreibend	doppelseitig	unmittelbar, in separatem Fenster	WinHelp, Apple Help Unix man-Pages
Tutorien	Nutzer trifft Auswahl	Verzeichnis	beschreibend	wechselseitig	unmittelbar, in separatem Fenster	Tutorial, Guided Tour, Demonstration
Tipps beim Anwendungsstart	Keine	per Zufall	beschreibend	einseitig vom System	unmittelbar, in separatem Fenster	PowerArchiver, WinCvs, u.v.a.
Intelligente Hilfe	System analysiert Nutzungshistorie und aktuellen Kontext	nach Analyse	hinweisend	wechselseitig	unmittelbar, in separatem Fenster	Microsoft Office-Assistent
Nutzer- und Expertenmodellierung	System analysiert Nutzungshistorie	nach Analyse	beschreibend	einseitig vom System	verzögert, in separatem Fenster	OWL
Über-Sichten	Nutzer trifft Auswahl	Verzeichnis	hinweisend	doppelseitig	unmittelbar, in separatem Fenster	
Neutralmodus	Nutzer zeigt auf Interaktionselement	aktueller Fokus	unterstützend	doppelseitig	unmittelbar, in Fokusnähe	
Spieldaten	Nutzer zeigt auf Interaktionselement	nach Analyse	unterstützend	wechselseitig	unmittelbar, in Fokusnähe	
Explorationskarten	Nutzer trifft Auswahl	Verzeichnis	beschreibend	einseitig vom System	unmittelbar, in separatem Fenster	
Newsgroups und FAQs	Nutzer formuliert Frage	nach Analyse	beschreibend	wechselseitig	verzögert, außerhalb der Anwend.	
verzögerte intelligente Hilfe	System analysiert Nutzungshistorie	nach Analyse	beschreibend	einseitig vom System	verzögert, außerhalb der Anwend.	vgl. <i>maxblue</i> -Fallstudie (S. 43ff)
verzögerte Explorationskarten	System analysiert Nutzungshistorie	nach Analyse	beschreibend	einseitig vom System	verzögert, in separatem Fenster	vgl. <i>maxblue</i> -Fallstudie (S. 43ff)

Tab. 3—1: Lernkomponenten, Hilfe- und Assistenzsysteme

3.2. Online-Handbücher

Nutzer rufen Online-Handbücher mit einem spezifischen Ziel auf. Oft haben sie gerade Schwierigkeiten bei der Erledigung einer Aufgabe. Sie wissen, was sie möchten, aber nicht, wie sie dies erreichen. Aller Wahrscheinlichkeit nach sind sie irgendwie ungeduldig oder frustriert, weil sie ihr Ziel nicht selbständig erreichen konnten [Apple 2002].

Um die Antwort zu finden, können Nutzer auf ein Inhaltsverzeichnis der Einträge zugreifen. Je nach Ausführung des Handbuches stehen auch ein Schlagwortverzeichnis und eine Volltextsuche zur Verfügung. Antworten bzw. Lösungen bieten Online-Handbücher in Form von textuellen Beschreibungen, evtl. angereichert mit Illustrationen. Mehrere aufeinander folgende Anfragen durch einen Nutzer werden in keinerlei Beziehung gesetzt, so dass die Kommunikation doppelseitig ist.

Bei Online-Handbüchern ist die Hilfe zwar in das System integriert, aber der Nutzer muss seinen Arbeitsablauf unterbrechen, um sie zu erhalten [Marion 97]. Sie werden in einem separaten Fenster angezeigt, das oft Teile der Benutzungsschnittstelle verdeckt, zu denen der Nutzer Fragen hat [Lewis/Rieman 93]. Der Nutzer wird dann zu mehr oder weniger umständlichen Wechseln zwischen der Anwendung und der Hilfe genötigt. Moderne Anwendungen versuchen manchmal, dies Problem zu umgehen, indem der Bildschirm geteilt wird: An der rechten Seite wird dann ein schmales Fenster für die Online-Hilfe dargestellt, während links davon die Anwendung in einem angepassten Fenster gezeigt wird.

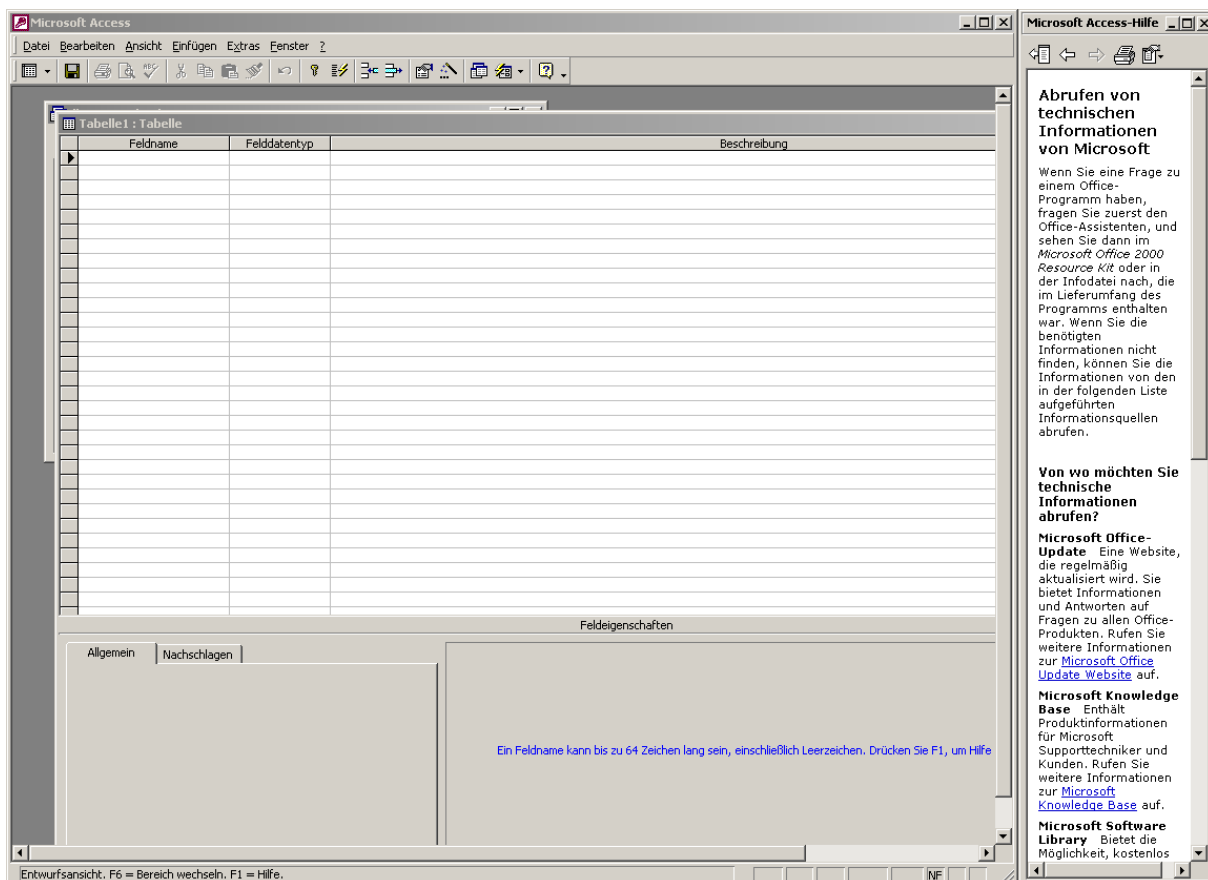


Abb. 3–2: Microsoft Access-Hilfe mit eingeklappten Registerkarten

Doch auch dies ist durchaus problematisch. In dem in Abb. 3–2 gezeigten Beispiel ist das Fenster für die Online-Hilfe standardmäßig so schmal, dass die Erklärungen schwer lesbar sind. Die Textseite wird sehr lang, so dass der Nutzer weit nach unten scrollen muss, um alles lesen zu können. Wenn der

Nutzer das Fenster der Hilfe verbreitert, überdeckt es wiederum einen großen Teil des Anwendungsfensters. Dies geschieht bei dem gezeigten Beispiel ebenso, wenn der Nutzer die Registerkarten für die Navigation in der Online-Hilfe ausklappt. Das Hilfefenster nimmt dann immerhin schon mehr als ein Drittel der Bildschirmbreite ein, aber viele Kapitelüberschriften im Inhaltsverzeichnis werden nicht vollständig angezeigt und der Textteil bleibt weiterhin sehr schmal (siehe Abb. 3–3).

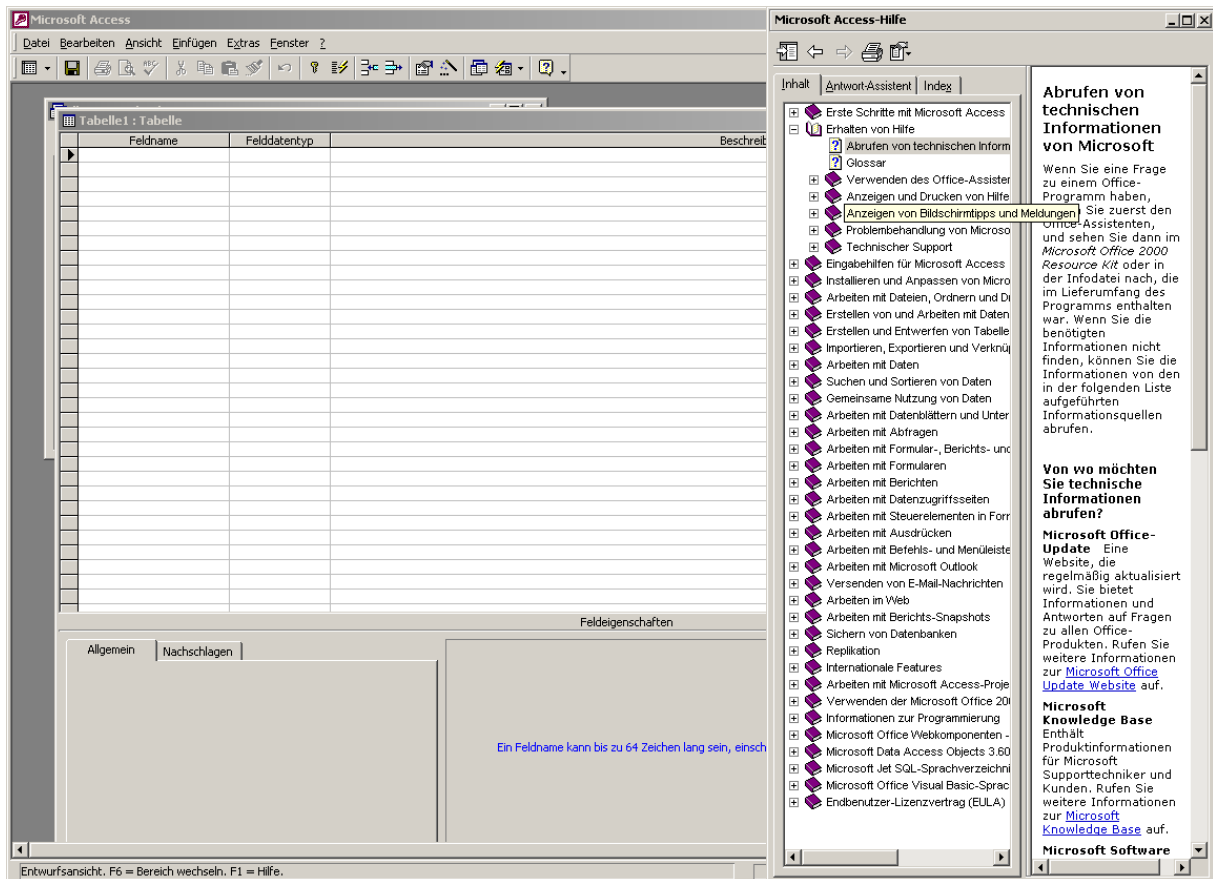


Abb. 3–3: Microsoft Access-Hilfe mit ausgeklappten Registerkarten

Beim Umgang mit Online-Handbüchern müssen die Nutzer lernen, Material zu selektieren, relevante von irrelevanten Informationen zu unterscheiden und den dargebotenen Text zu verstehen. Sie sind am besten für eine allgemeine Unterstützung bei gut verstandenen Problemen mit einfachen Lösungen geeignet [Covi/Ackerman 95].

Am häufigsten wird beklagt, dass Online-Handbücher zu viel Informationen anbieten. Besonders für Neulinge sind die Inhalte zu umfangreich. Dies führt dazu, dass Nutzer in der Masse der möglichen Hilfen und Dokumentation oft nicht die Information finden, die sie möchten, und selbst wenn sie sie finden, können sie die Hilfe fehlinterpretieren.

Außerdem bilden Online-Handbücher meist nur Programmfunktionen ab. Sie orientieren sich nicht an den Zielen der Nutzer und enthalten keine Erklärungen zum Hilfe-System selbst (vgl. [Nielsen 93a], [Lewis/Rieman 93], [Mirel 98], [Adams et al. 2001]).

Zur Erleichterung kontinuierlichen Weiterlernens sind Online-Handbücher nicht geeignet. Prinzipien des Minimalismus werden nicht berücksichtigt und die Erklärungen orientieren sich weder an den Aufgaben der Nutzer noch sind sie kurz gefasst. Exploration und intrinsische Motivation werden nicht genügend gefördert.

3.3. Tutorien

Bei Tutorien muss der Nutzer zum Aufrufen eine Auswahl aus einem Verzeichnis vorbereiteter Lektionen treffen. Die Inhalte einer Lektion werden beschreibend vermittelt. Die Kommunikation ist mindestens doppelseitig. Verfügt die Komponente auch über eine Fortschrittsanzeige oder markiert bereits durchgenommene Lektionen, so lässt sich die Kommunikation auch als wechselseitig bezeichnen. Tutorien laufen nicht direkt im Anwendungsfenster ab, sondern in einem eigenen.

Im Vergleich zu Online-Handbüchern behandeln Tutorien umfassendere Themen. Die Nutzer lernen etwas über Fähigkeiten und Aufgaben einer Anwendung, indem sie sich mit einer Kombination von Erklärungen, Beispielen und spielerischen Experimenten oder Übungen befassen [Selber/Johnson-Eilola/Mehlenbacher 97]. Die meisten Tutorien haben das Ziel, Transferleistungen anzuregen. Die Nutzer bearbeiten nicht ihre eigenen Aufgaben, sondern Standardaufgaben, die zu diesem Zweck vorbereitet wurden, um dann das Gelernte auf den eigenen Aufgabenbereich zu übertragen [Farkas 98]. Allgemein wird angenommen, dass sich Tutorien besonders für Neulinge eignen (z.B. [Encarnação 97], [Yom/Wilhelm 2002]). Entsprechend werden Tutorien meist für die anfängliche Lernphase konzipiert und behandeln grundlegende Konzepte, die beabsichtigte Verwendung und allgemeine, wichtige Funktionalitäten einer Anwendung [Wen 2000]. Doch prinzipiell spricht nichts dagegen, Tutorien auch für erfahrene Nutzer bereitzustellen. Den Erfolgsbeweis durch praktische Umsetzung hat Russell Brown mit seinen Anleitungen zu der Bildbearbeitung Adobe Photoshop, auf die hier im Abschnitt »Demonstrationen« eingegangen wird, eindrucksvoll erbracht.

Ich betrachte hier zwei wesentliche Ausprägungen von Tutorien: Demonstrationen (bzw. Guided Tours) und interaktive Tutorien.

Demonstrationen

Eine Demonstration ist die Wiedergabe von aufgezeichneten Nutzer-System-Interaktionen. Der Nutzer sieht in animierter Form, welche Aktionen er ausführen und welche Systemantworten er daraufhin erwarten kann.

Während klassische Dokumentations- und Hilfearten wie (reale oder elektronische) Handbücher eher beschreiben und erzählen, zeigen Demonstrationen Lösungswege und Verfahren in animierten Bildern. Gerade bei graphischen Benutzungsschnittstellen ist diese Art, Ereignisse darzustellen, einfacher und effektiver als die Erzählform [Rosson/Carroll 2002].

Hervorragende Beispiele dieser Art sind die bereits erwähnten »Tips & Techniques« zu Adobe Photoshop von Russell Brown (<http://www.russellbrown.com/tips/photoshop.html>, zuletzt besucht am 20.03.2003). Dies sind etwa zwei- bis zehnminütige Filme, in denen Themen wie das Verändern von Hintergründen, das Konvertieren von Farb- in Schwarz-Weiß-Bilder oder das Hinzufügen von Wasser-Effekten behandelt werden.

Die Filme sind Aufzeichnungen der Aktionen auf der Anwendungsoberfläche mit Audio-Kommentaren. Dort werden auf unterhaltsame Weise in kurzer Zeit relativ komplexe Verfahren beschrieben, wie man eine bestimmte Aufgabe in Adobe Photoshop am besten erledigen kann. Abbildung 3–4 zeigt ein Abspielprogramm, mit dem sich der Nutzer solch eine filmische Demonstration ansehen kann. Der Nutzer kann alle Cursorbewegungen, Eingaben und Reaktionen der Photoshop-Anwendung beobachten. Am unteren Rand des Abspiel Fensters befinden sich die Elemente, mit denen der Nutzer das Abspielen steuert. Er kann den Film jederzeit anhalten, vor- und zurückspulen, um die Abspielgeschwindigkeit an seine persönliche Aufnahmefähigkeit anzupassen. Diese Anleitungen sind im Gegensatz zu anderen Tutorien nicht für Neulinge gedacht. Es werden keine Grundlagen, sondern recht spezielle Funktionalitäten erklärt. Um davon profitieren zu können, muss man als Nutzer schon einige Erfahrung im Umgang mit Adobe Photoshop haben.

Als sinnvolle Einsatzmöglichkeiten für Demonstrationen zählt [Tidwell 99] auf:

- Es ist unklar, welcher Schritt als nächster folgt.
- Der Lösungsweg für eine Aufgabe ist einfach zu erinnern, sobald er einmal gezeigt wurde.
- Der Nutzer hat nicht die Zeit oder Lust, eine Anleitung zu lesen.
- Die Lösungsschritte sind schwer in Worte zu fassen.

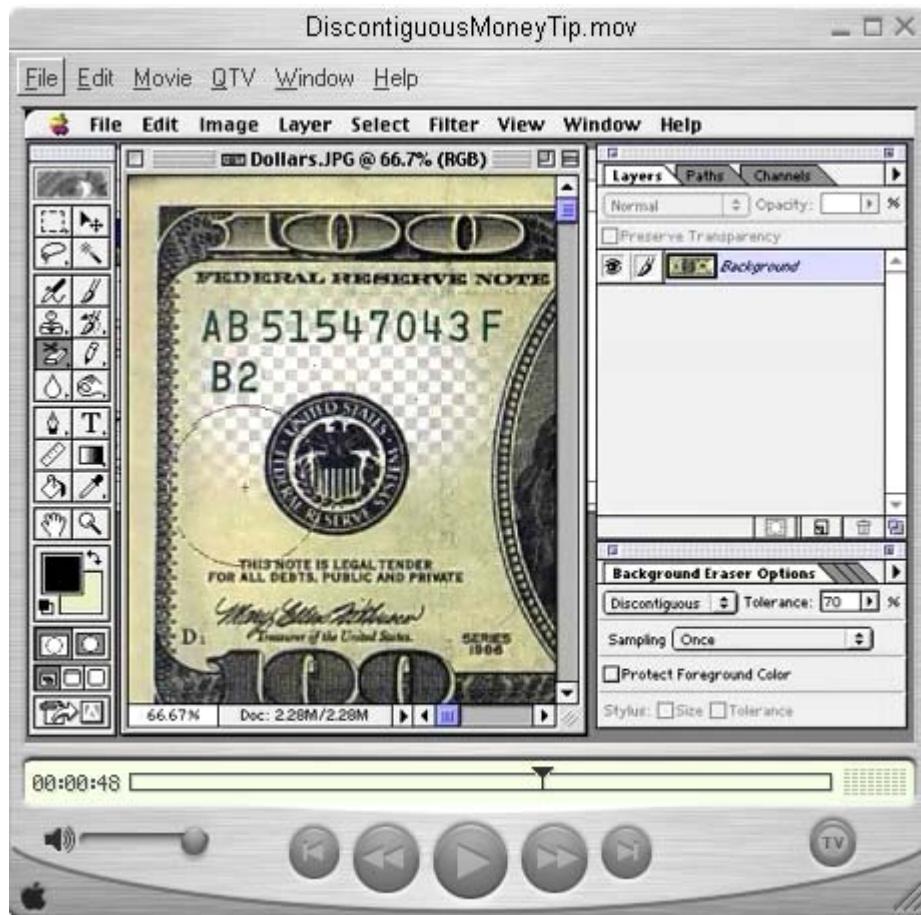


Abb. 3—4: Beispiel einer Photoshop-Anleitung von Russell Brown

Wenn der dargestellte Aufgabentyp keine feste Struktur hat, sollten mehrere Variationen demonstriert werden, um sowohl die gemeinsamen wie die speziellen Aspekte hervor zu heben. [Rosson/Carroll 2002] geben zu bedenken, dass Demonstrationen durch die Animation zwar Aufmerksamkeit erregen und sichere Vorschauen auf Funktionen gewähren, aber man durch bloßes Zusehen nicht lernen kann, wie man eine Funktion findet, ausführt und überwacht. Die Nutzer sollten also nicht nur zusehen, sondern zumindest den Ablauf der Demonstration steuern können, denn je höher der Interaktionsgrad für den Nutzer ist, desto höher ist auch der zu erwartende Lerneffekt. Gleichzeitig verringert sich dabei auch die Gefahr eines Abbruchs der Demonstration aus Langeweile [Gronover/Senger/Riempff 2002].

Häufig wird statt des Begriffes »Demonstration« auch die Bezeichnung »Guided Tour« verwendet (z.B. [Encarnaç o 97], [Yom/Wilhelm 2002]), wobei dann der Überblick über ein gesamtes System oder bestimmte Bereiche mehr im Vordergrund steht. Bezogen auf Anwendungen im Internet charakterisieren [Yom/Wilhelm 2002] eine Guided Tour als Rundgang durch ein Webangebot.

Interaktive Tutorien

Bei einem interaktiven Tutorium hat der Übungsaspekt eine größere Bedeutung. Der Nutzer hat hier mehr Einflussmöglichkeiten als bei einer reinen Demonstration, die durchaus Teil eines interaktiven Tutoriums sein kann. Bei einer Demonstration zeigt das System dem Nutzer das Bild eines Anwendungszustandes und überlässt es dann dem Nutzer, das System in diesen Zustand zu bringen. Bei einem interaktiven Tutorium erkennt das System selbst den Zustand und richtet daran die als nächstes angezeigten Informationen der Übung aus. Dies ist besonders hilfreich für das Fehler-Management. Interaktive Tutorien ermöglichen eine Unterstützung, die dynamisch an die Bedürfnisse und Aktivitäten von Nutzern gekoppelt ist, können Nutzer aber auch überfordern, wenn sie inhaltlich zu umfangreich sind, oder durch graphische Animationen, Videos und Tonausgaben eine Reizüberflutung verursachen [Rosson/Carroll 2002].

Tutorien eignen sich in gewissem Maße zur Erleichterung kontinuierlichen Weiterlernens, da sie prinzipiell aufgabenorientiert sind. Je mehr die vorbereiteten Standardaufgaben den realen Aufgaben der Nutzer entsprechen, desto bessere Lerneffekte können erzielt werden. Je nach Aufgabenstellung unterstützen Tutorien auch Prinzipien des Minimalismus und Exploration durch den Nutzer. Werden Demonstrationen so unterhaltsam wie von Russell Brown aufbereitet, fördern sie sicherlich intrinsische Motivation. Dies ist aber eher die Ausnahme als die Regel. Viele Online-Tutorien werden von Nutzern nicht verwendet, da sie sehr zeitaufwändig sind und meist nur geringe Praxisrelevanz besitzen [Encarnaço 97].

3.4. Tipps beim Anwendungsstart

Viele Anwendungen bieten den Nutzern an, beim Start einen Hinweis zur Bedienung zu geben. Diese Hinweise beziehen sich auf einzelne Funktionen, bestimmte Optionen oder auch auf Strategien zur Aufgabenbearbeitung.

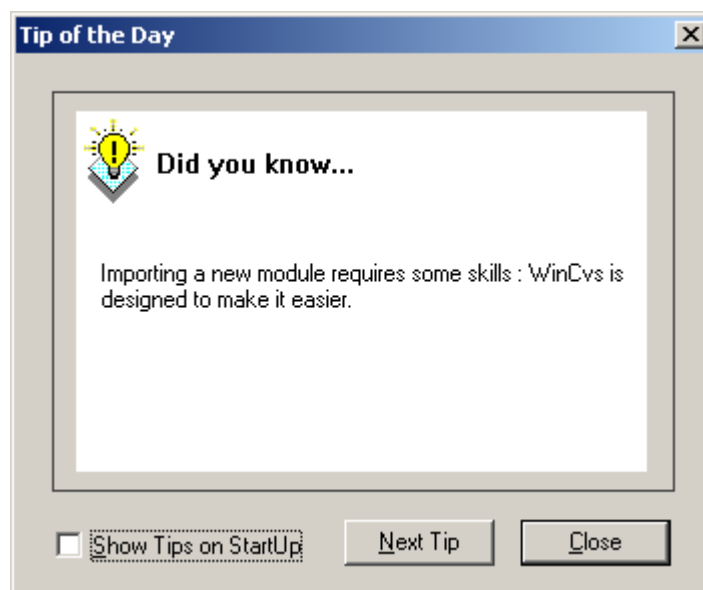


Abb. 3—5: Tipp beim Start des Versionskontrollsystems WinCvs

Da die Hinweise unmittelbar nach dem Programmstart ausgegeben werden, gibt es keine Möglichkeit, inhaltlich Bezug auf einen Anwendungskontext zu nehmen. Sie wenden sich nicht an eine spezielle Frage des Nutzers oder ein konkretes Problem, sondern werden rein zufällig »auf gut Glück« verteilt, so dass es unwahrscheinlich ist, dass sie eine effektive Hilfe bieten [Houser/DeLoach 98]. Viel mehr lenken sie den Nutzer, der das Programm zu einem bestimmten Zweck aufgerufen hat, ab und sorgen so für eine Behinderung. Der Nutzer hat keine Möglichkeit zu einer Reaktion – außer der, das Fenster mit dem Tipp zu schließen. Die Kommunikation des Systems verläuft also einseitig.

Die Hinweise sind zwar kurz gefasst und versuchen, den Nutzer zur Exploration anzuregen, aber sie stören viel zu sehr und berücksichtigen in keiner Weise die aktuellen Aufgaben des Nutzers. Zur Erleichterung kontinuierlichen Weiterlernens sind sie daher ungeeignet.

3.5. Intelligente Hilfe

Intelligente Hilfesysteme analysieren die aktuelle Nutzungshistorie und den jeweiligen Kontext, um so Probleme des Nutzers beim Umgang mit der Anwendung zu identifizieren. Werden solche Probleme erkannt, werden dem Nutzer Lösungen angeboten, die nicht nur eine Erklärung, sondern auch Verweise in die Anwendung beinhalten. So wird die Umsetzung eines Lösungsvorschlages und der

Wechsel vom separaten Hilfefenster zurück in die Anwendung erleichtert. Ein intelligentes Hilfesystem hat Zugriff auf die protokollierte Nutzungshistorie und kann so jede Eingabe und Reaktion des Nutzers berücksichtigen. Die Kommunikation ist daher wechselseitig.

[Nielsen 93b] erwähnt sowohl Chancen als auch Risiken aktiver Hilfesysteme. Denn obwohl aktive Hilfe einige Probleme traditioneller, passiver Hilfesysteme lösen könnte, so bestünde auch die große Gefahr, dass die Nutzer den Computer als aufdringlich und nörgelnd empfinden, wenn er sie bei ihrer Arbeit zu oft mit Ratschlägen unterbricht.

[Nielsen 93a] vermutet hinter den Anstrengungen zur Entwicklung intelligenter Hilfesysteme die »Erdnussbutter«-Theorie der Usability. Nach der bleibt der ursprüngliche Geschmack verborgen, wenn man etwas nur mit einer ausreichend dicken Schicht überdeckt. Intelligente Hilfe würde nach dieser Theorie vor allem eingesetzt, um ursprüngliche Probleme der Benutzungsschnittstelle zu überdecken, statt sie wirklich zu lösen. (Da im deutschen Sprachraum der Genuss von Erdnussbutter viel weniger verbreitet ist als in den USA, könnte man den Ausdruck vielleicht mit »Ketchup«-Theorie anpassen.) Intelligente Hilfe hätte allerdings insofern ihre Berechtigung, als dass Hilfesysteme tatsächlich hilfreicher sind, wenn sie erkennen können, was der Nutzer machen möchte und Ratschläge geben auf der Grundlage eines Modells darüber, was der Nutzer bereits weiß.

Genau hier liegt aber ein sehr grundlegendes Problem, denn nur aus den Aktionen einer Person kann nicht bestimmt werden, welche Ziele die Person gerade verfolgt [Rosson/Carroll 2002]. Auch [Lewis 98] schildert das Problem, dass Nutzer ihre Absichten dem System nur schwer mitteilen können und das System seine Antworten den Nutzern kaum verständlich machen kann.

[Rosson/Carroll 2002] beschreiben zwei weitere problematische Aspekte intelligenter Systeme. Sie besitzen die Stärke, im Aufgabenkontext des Nutzers zu agieren, wodurch sie spezielle Unterstützung bei bestimmten Aktionen und Ereignissen bieten können. Dabei haben sie aber die Schwäche, sich häufig zu irren und sind damit irreführend und können den Nutzer verärgern. Das Erstellen von Nutzermodellen zur Automatisierung oder Unterstützung von Aufgaben kann die Nutzung vereinfachen, kann aber den Eindruck der Steuerbarkeit des Systems reduzieren oder Bedenken gegen die Speicherung persönlicher Daten hervor rufen.

Die bekannteste Umsetzung intelligenter Hilfe ist sicherlich der Microsoft Office-Assistent. Aus diesem Grund wird an dieser Stelle ausführlich auf ihn eingegangen. Viele der oben aufgeführten Vor- und Nachteile lassen sich an diesem konkreten Beispiel nachvollziehen.

Microsoft Office-Assistent

Der Office-Assistent wurde von Microsoft in der Office '97 Produktfamilie eingeführt. Er ist ein animierter Charakter, der im Office '97-Paket als intelligenter Dreh- und Angelpunkt der Nutzer-Unterstützung fungieren sollte [Microsoft 96]. In der deutschsprachigen Version heißt sein Standardcharakter offiziell »Karl Klammer«, in der englischsprachigen »Clippit«, obwohl er unter seinem jeweiligen Spitznamen »Die Büroklammer« bzw. »Clippy« wesentlich bekannter sein dürfte.



Abb. 3—6: Microsoft Office-Assistent von 1997

Der Office-Assistent verfügt über zwei Interaktionsformen: eine passive und eine aktive. In der passiven, die in [Horvitz o.J.] »query based interaction« genannt wird, dient er als Schnittstelle zur normalen Online-Hilfe. Der Nutzer wendet sich an den Assistenten, um ihm ausformulierte (oder auch stichwortartige) Fragen zu stellen. Der Assistent analysiert die Eingabe und verweist dann auf entsprechende Texte in der Online-Hilfe. In der aktiven Interaktionsform, in [Horvitz o.J.] »event based interaction« genannt, beobachtet und analysiert der Assistent den Umgang des Nutzers mit der

Anwendung. Wenn er zu dem Ergebnis kommt, dass der Nutzer von einer Hilfestellung profitieren könnte, gibt er selbständig Tipps und Hinweise, die dem Nutzer den Umgang erleichtern sollen. Als berühmtes Beispiel sei hier angeführt, dass, sobald ein Nutzer ein neues Word-Dokument mit »Dear Mrs. Green« beginnt, der Office-Assistent folgert, dass der Nutzer einen Brief schreiben möchte, und die passenden Hinweise anzeigt [Reese Hedberg 98].

Die aktive Ausgabe der Tipps erfolgt dabei auf zwei Arten. Tipps, die als wichtig angesehen werden, werden unverzüglich in der Sprechblase des Assistenten angezeigt. Bei weniger wichtigen Tipps erscheint neben dem Assistenten eine Glühlampe, die anzeigen soll, dass der Assistent eine Idee hat, wie er dem Nutzer helfen kann. Der Nutzer muss selbst auf die Glühlampe klicken, wenn er den Tipp sehen möchte. Mit dieser proaktiven Unterstützung wurde das Ziel verfolgt, dem Nutzer das Gefühl zu geben, dass die Software für ihn arbeitet [Microsoft 96]. Inwieweit dieses Ziel erreicht wurde, wird am Ende dieses Abschnitts bewertet.

Historisch gesehen ist der Office-Assistent eine Weiterentwicklung des Antwort-Assistenten, den es bereits in Microsofts Office '95 gab. Dieser verfügte lediglich über die passive Interaktionsform, in der der Nutzer Fragen stellen konnte, die mit einer Zusammenstellung von Texten aus der Online-Hilfe beantwortet wurden [Heckerman/Horvitz 98]. Die Entwickler überlegten ursprünglich, den Office-Assistenten als eine alternative Benutzungsschnittstelle vergleichbar mit Microsoft Bob zu entwickeln (zu Microsoft Bob siehe Abschnitt 3.9 Anthropomorphe Schnittstellen, S. 39ff). Dieser Ansatz wurde jedoch aufgrund der Komplexität des Office Produktes nicht verwirklicht [Microsoft 96].

Der Antwort-Assistent und der Office-Assistent beruhen beide auf Ergebnissen des Lumière-Projektes der Microsoft-Forschungsabteilung. Die folgende Beschreibung dieses Projektes beruht v.a. auf [Horvitz et al. 98], [Horvitz 99a] und [Horvitz o.J.]. Eine Gruppe um Eric Horvitz hat in diesem Projekt, das 1993 gestartet wurde, Methoden und eine Architektur entwickelt, um auf die Ziele und Bedürfnisse von Nutzern bei ihrer Arbeit mit einer Software zu schließen. Der entwickelte Prototyp umfasste den Bereich der Anwendung Microsoft Excel.

Um auf die Ziele der Nutzer zu schließen, wurden relativ komplexe stochastische Verfahren (genauer gesagt Bayes-Modelle) verwendet. Für jeden potentiellen Hinweis wurde sein erwarteter Nutzen berechnet. Der erwartete Nutzen eines Hinweises wurde als Differenz aus erwartetem Gewinn und den erwarteten Kosten angesehen. Als Gewinn galten die möglichen Effizienzsteigerungen, die ein Nutzer nach Befolgen des Hinweises erfährt. Als Kostenfaktoren wurden Frustration und Ineffizienz gezählt, die sich aus der Ablenkung durch die Darstellung eines sonst inhaltlich wertvollen Hinweises oder aus der Befolgung eines unpassenden Hinweises ergeben hätten. Um solche Ablenkungen in wichtigen Situationen zu vermeiden, wurde versucht, aus den Interaktionen die Konzentration des Nutzers auf die aktuelle Aufgabe und seine Aufnahmefähigkeit zu erkennen. Bei der Entwicklung des Prototyps hatte man nämlich erkannt, dass ein ungeeigneter Ausgabezeitpunkt der Hinweise zu sehr hoher Frustration (also entsprechend hohen Kosten) beim Nutzer führen kann. Andererseits ist die Wahl des richtigen Zeitpunkts geeignet, das Erlebnis stark zu verbessern und dem Nutzer das Gefühl zu vermitteln, der Assistent versuche mit ihm zu arbeiten. Wie erwähnt gehörte dies zu den Kernzielen, die mit dem Office-Assistenten verfolgt wurden.

Erst wenn der so berechnete erwartete Nutzen über einem gewissen Schwellenwert lag, wurde der Hinweis tatsächlich ausgegeben. Diesen Schwellenwert wiederum konnte der Nutzer selbst auf einer Skala einstellen. Damit hatte der Nutzer die Kontrolle wie oft bzw. wie selten der Assistent mit einem Hinweis erscheinen sollte. Bei jeder Ausgabe eines Hinweises wurde auch diese Skala angezeigt, um den Schwellenwert so einfach wie möglich verstellen zu können.

Falls der Nutzer des Prototypen ein Hilfeangebot nicht wahrnahm (also nicht auf den Verweis in die Online-Hilfe klickte, sondern mit seiner Arbeit fortfuhr), so wurde nach einem Timeout im Titel des Hinweises eine Entschuldigung eingeblendet und weitere Hilfeangebote wurden zu diesem Thema nicht mehr gegeben.

Gleichzeitig wurde die gesamte Sitzung des Nutzers auf Schwächen bei seiner Interaktion untersucht. Am Ende der Sitzung wurde ihm dann eine Zusammenstellung von Hilfethemen präsentiert, die als besonders interessant für ihn beurteilt wurden. Diese Zusammenstellung war als Lektüre für eine ruhige Minute gedacht. In [Horvitz o.J.] findet sich die Überlegung, diese »Help backgrounder« genannte Zusammenstellung nicht am Ende der Sitzung auszugeben, sondern nur auf explizite Anfrage des Nutzers, dafür erweitert um eine Option zum Ausdrucken. Leider findet sich in den

verschiedenen Quellen über das Lumière-Projekt keine weitere Erwähnung über diese Hilfeform. Da sie später im Office-Assistenten auch nicht umgesetzt wurde, kann sie hier nicht weiter behandelt oder bewertet werden.

Der aus diesem Prototyp hervorgegangene Office-Assistent deckt ein wesentlich breiteres Themengebiet ab als der Lumière-Prototyp (nämlich alle Anwendungen des Office-Pakets), hat jedoch einen kleineren Funktionsumfang. Wie gesagt, wurde der »Help backgrounder« gar nicht umgesetzt, und es wurde auch auf die Entschuldigungen verzichtet, falls ein Nutzer einen Hinweis nicht wahrnimmt. Die Umsetzung mit verkleinertem Funktionsumfang lag u.a. in einer Konzentration eines erheblichen Teils der Ressourcen auf die Entwicklung von Internet-bezogenen Features der Office Anwendungen begründet [Reese Hedberg 98].

Auf Veranlassung des Office-Teams wurde dann auch noch der komplexe Ansatz mit den Bayes-Modellen um ein einfaches, regelbasiertes System »erweitert«, um den Assistenten mehr in den Vordergrund zu rücken. Die Forschungsabteilung, die den Lumière-Prototypen entwickelt hatte, äußerte Bedenken gegen diesen Ansatz. Sie befürchtete, dass die Nutzer durch dieses System zu sehr abgelenkt und damit gestört werden. Der Einsatz des regelbasierten Systems hatte nämlich zur Folge, dass die Hinweise inhaltlich nicht mehr so gut an die jeweilige Nutzungssituation angepasst waren, wie sie es noch im Prototypen waren. Es fiel auch die Berechnung der Schwellenwerte für den erwarteten Nutzen und die entsprechende Einstellungsmöglichkeit für die Nutzer weg. Hinweise wurden also häufiger angezeigt, was ja auch in der Intention des Office Teams lag. Den Nutzern blieb als Kontrolle nur der über die Menüs erreichbare Optionen-Dialog.

[Houser/Deloach 98] formulieren als Anforderungen an eine Unterstützung für Nutzer (wohlgemerkt in dieser Reihenfolge!), dass sie unaufdringlich sein soll, zum angemessenen Zeitpunkt erscheinen soll und den passenden Inhalt darstellen soll. In [Horvitz 99b] werden die Problemfelder ganz ähnlich beschrieben. Wie man aus dem Vergleich von oben sieht, erfüllt der Office-Assistent diese Kriterien weit weniger als es der Lumière-Prototyp noch tat. Zur weiteren Bewertung des Office-Assistenten wird vor allem auf Ergebnisse von John Carroll und Heike Schaumburg zurückgegriffen.

John Carroll hat das Konzept intelligenter Hilfe durch einen Assistenten bereits 1988 in einer Studie untersucht [Carroll/Aaronson 88]. Carrolls System hieß »SmartHelp« und war ein Wizard-of-Oz-Prototyp, dessen Funktionalität der aktiven Komponente des Office-Assistenten entspricht. SmartHelp war ein aktives Hilfesystem, das ohne Anforderung des Nutzers Hilfeinformationen anzeigte, wenn der Nutzer einen Fehler gemacht hat oder bei der Anwendung nicht mehr weiter wusste. Die Meldungen von SmartHelp erschienen in einem eigenen Fenster mit einem akustischen Signal. Bereits 1988 kam Carroll so mit recht einfachen Mitteln zu Ergebnissen, die aufgrund der Ähnlichkeit der Systeme sehr gut auf den Lumière-Prototypen und besonders auf den Office-Assistenten übertragbar sind. In problematischen Anwendungssituationen wurde SmartHelp von einigen Nutzern tatsächlich als Begleitung empfunden, die ermutigen und bestätigen konnte. Neben diesen hilfreichen Episoden wurden aber auch Situationen beobachtet, in denen SmartHelp keine oder nur ungeschickte Hilfe lieferte oder sogar selber die Quelle für weitere Probleme des Nutzers war. Die Meldungen von SmartHelp wurden oft als Ablenkung empfunden und daraufhin ignoriert. Die Nutzer fühlten sich frustriert, wenn Tipps zu spät kamen, nachdem sie selber eine Lösung für ein Problem gefunden hatten. Die Forscher fühlten sich von der Erkenntnis entmutigt, wie dünn die Grenze zwischen Verärgerung und Unterstützung der Nutzer war. Erfuhren die Nutzer einmal ein Problem bei der Arbeit mit SmartHelp, und sei es nur wegen eines kleinen Missverständnisses, stellten einige die Kompetenz und Glaubwürdigkeit des Systems überhaupt in Frage und beschuldigten pauschal das Hilfe-System für alle weiteren auftretenden Probleme. Intelligente Systeme dieser Art wurden daher nicht nur als Lösungsansätze, sondern gleichzeitig auch als Quelle von Problemen im Bereich der Gebrauchstauglichkeit bewertet. Dies wurde als ein sehr hoher Kostenfaktor für die Vermittlung von Intelligenz in einer Anwendung gesehen. Die Autoren verweisen auch auf die Erkenntnis, dass gute Trainer nicht jedes Mal eingreifen, wenn sie selbst es besser machen könnten. Vielmehr läge ihr Erfolg in der Kunst der Zurückhaltung, die den Nutzer dann auch eigene Wege finden lässt. Wie oben beschrieben, hatte sich das Office Team von Microsoft gerade gegen so eine zurückhaltende Strategie entschieden.

Auch [Schaumburg 2001] sieht das Hauptproblem aktiver Softwarekomponenten in der Gefahr, die Nutzer bei ihrer Arbeit zu unterbrechen, da die Nutzer sich dann leicht gestört und abgelenkt fühlen und diese Komponente abschalten möchten. Die starke Abneigung gegen den Office-Assistenten, die auch viele Leser bei sich oder Bekannten wahrgenommen haben mögen, wurde in dieser Studie bestätigt, denn viele der Testpersonen wehrten sich dagegen, ihn zu benutzen. Der Grund wird darin gesehen, dass sich der Office-Assistent zwischen die Nutzer und deren Aufgaben stellt. Selbst bei einer anfänglich positiven Einstellung gegenüber dem Office-Assistenten, wird er von Nutzern abgelehnt, sobald sie sich einmal bei ihrer Aufgabenerfüllung durch ihn behindert fühlen.

In dieser Studie wurde auch speziell die passive Komponente des Office-Assistenten, also die Schnittstelle zur Online-Hilfe, untersucht. Die Mehrheit der Testpersonen gab an, dass sie den Office-Assistenten hierfür zwar als einigermaßen nützlich ansehen, seine Verwendung aber trotzdem ablehnen. Die Antworten zeigten, dass eine Frage an den Office-Assistenten als am wenigsten effiziente Methode beurteilt wird. Die Mehrheit der Testpersonen war viel eher geneigt, zu versuchen, das Problem selbst zu lösen, den Index der Online-Hilfe zu verwenden oder sich an einen erfahrenen Bekannten (wie einen lokalen Experten) zu wenden.

In der Studie wurden die fünf Variablen Ablenkung, Vertrauen, Verständnis, Vorhersagbarkeit und Kontrolle in ihrer Bedeutung für die Akzeptanz und Verwendung des Office-Assistenten untersucht. Das Ergebnis war, dass zwar alle fünf Variablen mit der Akzeptanz und Häufigkeit der Verwendung korrelierten, diese Korrelation jedoch nur für die Ablenkung und das Vertrauen signifikant waren. Man kann also folgern, dass der Office-Assistent nicht akzeptiert und verwendet wurde, da die Nutzer ihm vor allem nicht vertraut haben und sie bei ihrer Arbeit abgelenkt wurden. Als Einschränkung für die Gültigkeit der Ergebnisse wird in der Studie genannt, dass die Mehrheit der Teilnehmer viel Erfahrung mit Computern im allgemeinen und Microsoft Office im speziellen hatte. Möglicherweise hätten sich bei Neulingen andere Ergebnisse gezeigt. Außerdem beruhen die Ergebnisse nur auf den Selbstausskünften der Testpersonen.

Die zahlreichen Schwächen des Office-Assistenten, die hier dargestellt wurden, spiegeln sich auch in den durchweg negativen Beurteilungen wider, von denen hier stellvertretend nur zwei angeführt werden. So schreiben z.B. [Rosson/Carroll 2002]: »the infamous paper clip in Microsoft Office is another generally recognized example of technology overstepping its usefulness« [ebd., S. 82]. Später wird dort der Office-Assistent als übereifrig beschrieben, da er zu häufig oder auf unerwünschte Weise Hilfe anbietet. [Rupp 2001] möchte die Meinung der Nutzer zusammenfassen, wenn sie wie zu Beginn des Kapitels zitiert über diesen »nervigen kleinen ›Helfer« schreibt.

Welchen Umfang die Kritik von Wissenschaftlern bis hin zur Tagespresse hatte, mag man daran erkennen, dass man bei Microsoft als Antwort ein eigenes Werbebudget für die Botschaft geschaffen hat, dass die aktuelle Version des Office-Paketes ohne »die Büroklammer« auskommt. Teil der Werbekampagne ist eine Web-Site (<http://www.microsoft.com/Office/clippy/default.asp>, 31.10.2002), die sich ganz diesem Thema widmet. Dort wurde dem Office-Assistenten – von Microsoft selbst – u.a. auch der Spitzname »Mr. Annoying« verliehen.

3.6. Nutzer- und Expertenmodellierung

In [Linton/Joy/Schaefer 99] und [Linton et al. 2000] wird das OWL-System (OWL steht für »Organization-Wide Learning«) beschrieben, das basierend auf einem speziellen Nutzer- und Expertenmodell individualisierte Zusammenstellungen von Hinweisen ausgibt. Der Grundgedanke ist, dass man sich bei der Auswahl von Lernempfehlungen daran orientieren kann, was sich bei der Mehrheit anderer Nutzer bereits im praktischen Umgang bewährt hat. Ähnlich wie ein intelligentes Hilfesystem wird auch für die Nutzer- und Expertenmodellierung die Nutzungshistorie analysiert. Da die Ausgabe des OWL-Systems allerdings nicht unmittelbar sondern verzögert erst auf Anfrage des Nutzers erfolgt, muss hier nicht der jeweils aktuelle Kontext berücksichtigt werden.

Es handelt sich um einen statistischen Ansatz, bei dem sich das Nutzermodell aus Langzeitbeobachtungen von individuellen Verhaltensweisen ergibt und das Expertenmodell durch eine Zusammenfassung dieser entsteht. In der Praxis wurde das OWL-System für die Textverarbeitung Microsoft Word eingesetzt.

Für die Erstellung der Nutzermodelle wird jede Verwendung eines Kommandos individuell aufgezeichnet. Bei der Auswertung dieser Verhaltensweisen über einen längeren Zeitraum ergeben sich Muster von Verwendungshäufigkeiten für verschiedene Kommandos, nach denen auch eine Einteilung in Nutzergruppen stattfindet. Das Nutzermodell ist einfach die Liste der verschiedenen aufgezeichneten Kommandos zusammen mit der jeweiligen Nutzungshäufigkeit. Kommandos, die die meisten Nutzer einer Gruppe besonders oft verwenden, werden als besonders nützlich bewertet. Für das Expertenmodell werden die Modelle aus einer Nutzergruppe zusammengefasst und Durchschnittswerte berechnet. Ein Experte wird als fiktive Person angenommen, deren Nutzermodell konsistent zu allen realen Modellen der jeweiligen Nutzergruppe ist. Sowohl die individuellen Nutzermodelle wie auch das Expertenmodell werden regelmäßig aktualisiert, um Änderungen mitzubekommen, die sich mit der Zeit ergeben.

Die Hinweise, die ein Nutzer bekommt, werden durch den Vergleich der individuellen Verhaltensweisen mit dem Expertenmodell bestimmt. Dazu wird berechnet, ob ein spezielles Kommando von einem Nutzer gar nicht, seltener oder häufiger verwendet wird, oder ob es nahe an der Grenze zu einem dieser drei Bereiche liegt. Dem liegen folgende Überlegungen zu Grunde:

- Wird ein Kommando gar nicht verwendet, das die anderen Nutzer in der Gruppe offenbar nützlich finden, würde dieses Kommando angewendet, wenn der Nutzer es lernte.
- Wird ein Kommando seltener verwendet, kann dies ein Hinweis darauf sein, dass der Nutzer bereit ist, andere Anwendungsarten dieses Kommandos zu lernen.
- Wird ein Kommando häufiger verwendet, könnte dies ein simples Allzweck-Kommando (z.B. »Löschen«) anzeigen, an dessen Stelle ein mächtigeres Spezial-Kommando (z.B. »WortLöschen«) passender wäre.

Die daraus resultierenden Lernempfehlungen können z.B. nach der berechneten Nützlichkeit der jeweiligen Kommandos oder in der Reihenfolge der erwähnten Bereiche priorisiert werden. Die aktuelle Liste der Lernempfehlungen kann ein Nutzer jederzeit aufrufen. Zu jedem Kommando gibt es eine kurze Beschreibung und einen Knopf, mit dem die entsprechende Seite der normalen Online-Hilfe aufgerufen wird. Die Ausgabe enthält also lediglich Verweise zu bestehenden Beschreibungen der Kommandos. Die Kommunikation ist einseitig, da der Nutzer nicht aktiv auf die Lernempfehlungen reagieren kann. Er kann sich nur mit der Online-Hilfe auseinandersetzen und sein Nutzermodell nur mittelbar und unbewusst durch Anwendung des Gelernten beeinflussen.

Mit diesem System wurde erkannt, dass erfahrene Nutzer nicht komplett andere Kommandos verwenden als Nutzer mit weniger Erfahrung. Statt dessen verwenden sie einfach ein größeres Repertoire an Kommandos. Dabei ändern sich die jeweils nützlichsten Features einer Applikation mit der Zeit analog zum Wissens-Zuwachs (des Einzelnen und der Gruppe) und bei neuen Tätigkeitsfeldern. Die Autoren gestehen die Schwäche ein, dass dieser Ansatz sehr stark system- statt aufgabenorientiert ist, halten ihn aber dennoch für effizient, da seine Implementierung sehr einfach ist.

3.7. Explorationsunterstützung

Wie im Kapitel über lernpsychologische Grundlagen (Abschnitt 2.2, S. 6ff) dargestellt wurde, wird bei explorativer Interaktion zwischen Erkundung und Experimentieren unterschieden. Für diese beiden Interaktionsformen gibt es jeweils spezielle Unterstützungsarten.

Erkundung wird unterstützt durch:

- Über-Sichten
- Interaktionsgraphen
- Filter
- Neutralmodus
- Szenario-Maschine

Unterstützungsformen für das Experimentieren sind:

- Stornierung
- Wiederaufsetzpunkte
- Spieldaten
- Explorationskarten
- anpassbare Systeme

An dieser Stelle werden v.a. die Ansätze der Explorationsunterstützung besprochen, die im Zusammenhang mit der am Anfang dieses Kapitels vorgestellten Klassifikation (siehe S. 19ff) besonders interessant sind. Eine umfassendere Betrachtung zur Gestaltung explorationsfreundlicher Systeme findet der interessierte Leser in der hier schon mehrfach angeführten Dissertation [Paul 93]. Da die Unterstützung von Exploration allgemein sehr gut für eine Erleichterung kontinuierlichen Weiterlernens geeignet ist, gilt dies auch für die hier aufgezählten Ansätze.

3.7.1. Über-Sichten

Der Begriff »Über-Sichten« steht für Ansichten, die einen Überblick verschaffen sollen. Gemeint ist hier ein Überblick über bestimmte Systemfunktionen. In diesen Ansichten werden die jeweiligen Funktionen einerseits erklärt, andererseits können sie von dort auch direkt aufgerufen werden. Im Gegensatz zu klassischen Hilfe-Systemen findet hier kein Bruch zwischen dem Finden von Informationen und der darauf folgenden Anwendung statt. Die Antworten haben also hinweisenden Charakter.

Die Zusammenstellung von Funktionen findet nach verschiedenen Kriterien statt. Es sind sowohl Gruppierungen nach aufgabenorientierten Aspekten als auch unterschiedlich sortierte Referenzlisten aller Systemfunktionen möglich.

Der Nutzer muss eine Über-Sicht aus einer Liste auswählen. Die Kommunikation ist doppelseitig, da sich die Systemreaktionen nur auf die unmittelbar vorangegangenen Nutzeraktionen beziehen. Eine Über-Sicht wird in einem separaten Fenster angezeigt.

3.7.2. Neutralmodus

Der Neutralmodus ist eine spezielle Betriebsart eines Systems, in der dessen Funktionen nicht ausgeführt werden, sondern an ihrer Stelle kurze Funktionsbeschreibungen angezeigt werden. So stellt der Neutralmodus eine besondere Einladung zum Ausprobieren und Experimentieren dar. Da die Systemfunktionen nicht ausgeführt werden, vermittelt der Neutralmodus große Sicherheit und wird auch als »Konjunktivmodus« bezeichnet.

Seinen Informationsbedarf macht der Nutzer durch die simulierte Verwendung eines Interaktionselements kenntlich. Genau dort beim aktuellen Fokus des Nutzers antwortet das System sofort mit einer Erklärung der Auswirkungen, die sich im Realbetrieb ergäben. Die Antwort ist unterstützend, da der Nutzer das System im Neutralmodus genau so bedient wie später zur tatsächlichen Funktionsausführung. Die Kommunikation ist doppelseitig, da die verschiedenen Nutzereingaben ja eben keine Auswirkung auf das System haben, sondern nur mit Erklärungen beantwortet werden.

3.7.3. Spieldaten

Ebenso wie beim Neutralmodus ist der Nutzer auch bei der Arbeit mit Spieldaten sicher vor gefährlichen Auswirkungen unbekannter Funktionen. Diese werden zwar ausgeführt, bearbeiten dabei aber nur Daten, die speziell für diese Erkundungsart bereitgestellt wurden. Die Daten wurden vorbereitet, um die Wirkung der verschiedenen Funktionen besonders hervorzuheben. Die Auswahl der Daten in der Vorbereitung ist dabei durchaus nicht trivial, da sie weder zu spezifisch noch zu unspezifisch und für den Nutzer leicht verständlich sein müssen.

Die Funktionen, die der Nutzer aufruft, werden tatsächlich ausgeführt. Die Spieldaten unterstützen den Nutzer also dabei, die Auswirkungen der Aufrufe zu testen. Jeder Funktionsaufruf bearbeitet die Daten, so dass die Kommunikation zwischen System und Nutzer wechselseitig ist.

3.7.4. Explorationskarten

Explorationskarten enthalten Vorschläge für Experimente mit Systemfunktionen. Sie beruhen auf den »Guided exploration cards« aus [Carroll 90]. Die Karten haben jeweils vier Felder:

- In dem ersten Feld wird die Arbeitsaufgabe beschrieben. So erkennt der Nutzer gleich, ob der Inhalt für ihn interessant ist oder nicht.
- Das zweite Feld enthält die Hinweise, wie die gestellte Aufgabe zu lösen ist. Hier werden die eigentlichen Erklärungen zur Bedienung des Systems gegeben.
- Das dritte Feld dient dem Nutzer zur Erfolgskontrolle. Es beschreibt, woran man erkennt, dass man die Aufgabe erfolgreich gelöst hat. Der Nutzer erkennt daran ebenso, wann er eine Aufgabe noch nicht beendet hat.
- Im letzten Feld stehen Abhilfen für häufige Fehler bei der Aufgabenbearbeitung.

Explorationskarten passen gut zum minimalistischen Lernansatz, da die gegebenen Erklärungen unvollständig sind und sich die Inhalte an den Aufgaben der Nutzer orientieren. Der Nutzer muss aktiv mit dem System arbeiten, um die Erklärungen wirklich verstehen zu können. Da sie zusätzlich sehr kompakt sind, eignen sie sich besonders für lernentwöhnte Nutzer.

Dem Nutzer steht in der Anwendung eine Liste mit Explorationskarten zur Verfügung aus der er seine Auswahl treffen muss. Eine Karte mit einer Funktionsbeschreibung wird dann in einem neuen Fenster angezeigt. Genau wie bei den Tipps beim Anwendungsstart (S. 28) kann der Nutzer die Karte nur lesen und am Ende schließen. Während des Experimentierens, wie es vorgeschlagen wurde, ändert sich die Karte nicht. Die Kommunikation verläuft daher einseitig vom System.

3.8. Newsgroups und FAQs

Alle in den vorangegangenen Abschnitten behandelten Unterstützungsformen sind jeweils Bestandteil der Anwendung selbst. Aber einen großen Teil der Information über Software erhält man nicht im direkten Umgang mit ihr, sondern von anderen Menschen. Menschen helfen einander gern und mögen es, wenn jemand anderes ihren eigenen Informationsbedürfnissen seine persönliche Aufmerksamkeit schenkt. Wie effektiv die Hilfe anderer sein kann, erkennt man daran, dass ein lokaler Experte als die beste Unterstützungsform für einen Nutzer angesehen wird (vgl. Abschnitt 2.1 Der Umgang mit komplexen Systemen, S. 5f). Ist so ein lokaler Experte nicht verfügbar, kann man sich als Ersatz an Newsgroups wenden. Diese werden wegen der sozialen Mechanismen in diesen Gemeinschaften ebenfalls als gute Unterstützungsform für Nutzer angesehen (vgl. [Rosson/Carroll 2002], [Wen 2000]).

Eine Newsgroup ist eine strukturierte Diskussion mittels E-Mails, die über News-Server weltweit verbreitet wird. Beiträge werden zusammen mit dazugehörigen Antworten dargestellt, wodurch Diskussionsstränge entstehen [Rosson/Carroll 2002]. Nutzer formulieren ihre Fragen aus und senden sie an die anderen Teilnehmer der jeweiligen Newsgroup. Diese »analysieren« die Frage, um daraufhin in der Antwort ihre Lösungsvorschläge zu beschreiben. Ohne großen Aufwand können dort in einem wechselseitigen Dialog die Hilfebedürfnisse von Nutzern erläutert und Unklarheiten von Fragen, Antworten oder Lösungsvorschlägen beseitigt werden [Covi/Ackerman 95]. Durch die Speicherung der Diskussionsstränge werden sie Teil eines Informationsarchivs, das jedem zugänglich ist.

Viele Diskussionen in Newsgroups weisen ganz verschiedenartige Inhalte auf. Beispielsweise werden sowohl Geschichten von persönlichen Erfahrungen erzählt als auch detaillierte technische Beschreibungen gegeben oder einfach Fragen gestellt. Oft gibt es Beiträge, die immer wieder umfangreiche Meta-Diskussionen über Relevanz und Irrelevanz von Beiträgen anregen.

In [Rosson/Carroll 2002] wird auch der typische Lebenszyklus einer Newsgroup beschrieben: Am Anfang tauscht sich eine Expertengruppe aus z.B. Programmierern mit Spaß und zum Nutzen aller in einer Newsgroup aus. Es ist recht wahrscheinlich, dass die Qualität ihrer Diskussionen weniger erfahrene Programmierer anzieht. Anfängerfragen könnten dann Diskussionen verwässern und die Experten vertreiben, so dass das Niveau in der Newsgroup sinkt. Allgemein hat man meist schon Probleme, eine relevante Newsgroup und dort die passende Diskussion zu einer Frage überhaupt erstmal zu finden. Und selbst wenn man dies geschafft hat, ist es keineswegs sicher, dass jemand auf eine spezielle Frage antwortet, oder es besteht die Gefahr, dass wohlgemeinte aber fehlerhafte oder unvollständige Hilfe geboten wird. Nach [Covi/Ackerman 95] ist diese Gefahr aber immerhin geringer als bei maschinellen Hilfesystemen, weil die menschlichen Problemlösungs-Fähigkeiten den maschinellen überlegen seien.

Bei E-Mail-Diskussionen ist die zwischenmenschliche Distanz größer als in der Kommunikation von Angesicht zu Angesicht, wodurch Hemmungen dort geringer sind. Dies hat die Vorteile, dass z.B. persönliche Probleme oder unangenehme Wahrheiten leichter ausgesprochen und riskante Entscheidungen leichter diskutiert werden können. Andererseits folgen daraus auch Nachteile, denn unangemessenes bis unverschämtes Verhalten wird ebenso erleichtert. Das kann Konsensfindungen erheblich erschweren und sogar psychologischen Schaden erzeugen [Rosson/Carroll 2002].

FAQs

Die unbearbeiteten Informationsarchive der Newsgroups werden oft in FAQs (»frequently asked questions«, auf deutsch »häufig gestellte Fragen«) zusammengefasst. Diese Sammlungen immer wiederkehrender Fragen sollen allzu häufigen Wiederholungen in den Newsgroups entgegen wirken. Man kann in ihnen blättern und meist auch per Schlagwortverzeichnis auf sie zugreifen [Rosson/Carroll 2002].

So kann man allerdings aus der wechselseitigen Kommunikation nicht mehr direkt Vorteile ziehen. Stattdessen müssen die Hilfesuchenden die Interaktion zwischen den anderen Hilfesuchenden und den Antwortenden selbst nachvollziehen, um die Hilfeinformationen wieder verwenden zu können [Covi/Ackerman 95].

3.9. Anthropomorphe Schnittstellen

Anthropomorphe Charaktere sind Verkörperungen von Software-Agenten, die in größerem oder kleinerem Maße mit verschiedenen lebensähnlichen Elementen ausgestattet sind, um die Nutzer zu einer lebensechten Interaktion mit dem Agenten anzuregen [Dietz/Lang 99]. Sie sind ein zusätzlicher Kanal zur Kommunikation zwischen Nutzer und System. Aus diesem Grunde werden sie hier nicht als ein weiterer Ansatz in der Klassifikation von Lernkomponenten, Hilfe- und Assistenzsystemen beschrieben. Vielmehr können anthropomorphe Schnittstellen eine Bereicherung für viele der beschriebenen Ansätze sein. Sie werden als neue Schnittstelle zu Inhalten von Online-Handbüchern (vgl. S. 24f) eingesetzt, dienen als Führer in Guided Tours (vgl. S. 26ff) oder antworten flexibel und intelligent auf Fragen der Nutzer [Yom/Wilhelm 2002].

Wenn anthropomorphe Figuren u.a. auch als virtuelle Helfer bezeichnet werden [Krämer/Nitschke 2002], wird die Hoffnung deutlich, mit anthropomorphen Schnittstellen einen bestmöglichen Ersatz für (reale) lokale Experten anzubieten. Es gibt also vielfältige Beziehungen zwischen anthropomorphen Schnittstellen und anderen Formen der Lernunterstützung, die bis jetzt behandelt wurden.

Im Web findet man mittlerweile zahlreiche Ausführungen von anthropomorphen Schnittstellen, in denen die Charaktere entweder durch Bilder realer Personen oder durch computeranimierte Figuren dargestellt werden [Gronover/Senger/Riempff 2002]. Letztere verfügen teilweise über geradezu »natürlich anmutende Sprachausgaben und Bewegungen« [Krämer/Bente 2002, S. 206].

Aber nicht nur in modernen Web-Anwendungen sondern auch in klassischen PC-Applikationen findet man Formen von anthropomorphen Schnittstellen. So verfügte ja beispielsweise auch der Microsoft Office-Assistent über eine Ein- und Ausgabe komplett ausformulierter Sätze und wurde durch einen animierten Charakter dargestellt, weswegen er den anthropomorphen Schnittstellen zugerechnet wird

(vgl. [Dietz/Lang 99], [Schaumburg 2001]), auch wenn dieser zu seiner Entstehungszeit noch viel mehr von eingeschränkten technischen Möglichkeiten betroffen war als moderne anthropomorphe Schnittstellen. Einen Baustein für Charaktere dieser Art war das »MS Agent package« mit dem man die Elemente der Spracherkennung und -synthese sowie der Animation selbst umsetzen konnte [Horvitz/Paek 99].



Abb. 3—7: Aisa, ein Assistent für die Konfiguration von smart Autos

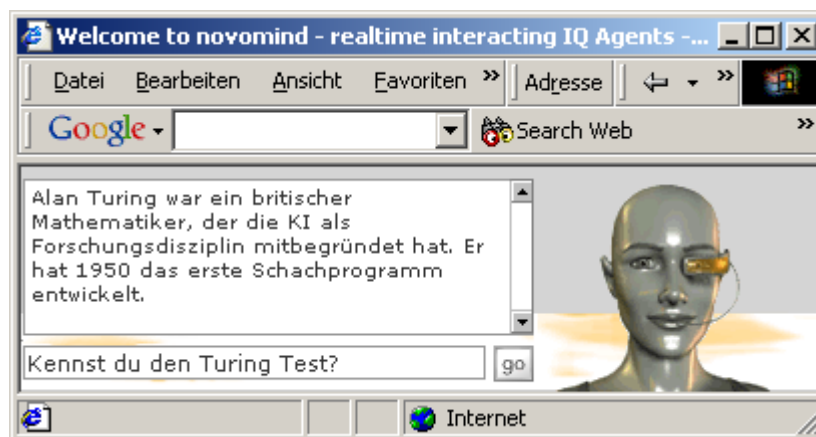


Abb. 3—8: Nomi, die virtuelle Repräsentantin der novomind AG

So beeindruckend einige dieser Beispiele auf den ersten Blick auch erscheinen mögen – wie vielleicht »Fritz Felge« von <http://www.reifen.com> (24.03.2003), den nicht einmal die klassische Smalltalk-Frage nach dem Wetter in Verlegenheit bringt – so verfügt jedoch keiner »über kommunikative Fähigkeiten, die denen eines menschlichen Gesprächspartners auch nur annähernd vergleichbar wären.« [Krämer/Bente 2002, S. 208]

In anthropomorphen Schnittstellen können natürlichsprachliche Ein- und Ausgaben in gesprochener oder geschriebener Form mit verschiedenen nonverbalen Kommunikationskanälen kombiniert werden. Durch die Animation der Figuren können Körperhaltung und -bewegung, Mimik, Gestik und Blickverhalten dargestellt werden, weswegen anthropomorphe Schnittstellen als multimodal bezeichnet werden. [Bente/Krämer 2000], [Krämer/Bente 2002] und [Krämer et al. 2002] führen zahlreiche Beispiele an, in denen anthropomorphe Schnittstellen wegen dieser Parallelen zwischen der Interaktion mit ihnen und menschlicher Face-to-Face-Kommunikation sehr oft per se als vorteilhaft

angesehen werden. Häufig werden diesen Schnittstellen a priori Vorzüge wie bessere Bedienbarkeit oder Steigerungen bezüglich Effizienz und Akzeptanz zugeschrieben. Dabei werden jedoch nur die Chancen gesehen, die sich aus der Erweiterung der Kommunikationskanäle gegenüber klassischen Benutzungsschnittstellen ergeben, ohne die damit gleichzeitig verbundenen Risiken zu berücksichtigen.

Eine realistische Abwägung zwischen den Vor- und Nachteilen anthropomorpher Schnittstellen ist noch nicht möglich, da der aktuelle Stand der Forschung diesbezüglich noch lückenhaft und unsystematisch ist (vgl. [Krämer/Bente 2001], [Schaumburg 2001]). Es soll dennoch kurz zusammengefasst werden, was bisher an Erkenntnissen gesammelt wurde. Ich greife dabei auf die Aussagen von [Bente/Krämer 2000], [Bente/Krämer 2001], [Dietz/Lang 99], [Gronover/Senger/Riempp 2002], [Krämer/Bente 2001], [Krämer/Bente 2002], [Krämer/Nitschke 2002], [Krämer et al. 2002] und [Rosson/Carroll 2002] zurück.

An erster Stelle sind hier die Ergebnisse von Byron Reeves und Clifford Nass (vgl. [Reeves/Nass 96]) zu nennen, die teilweise recht kontrovers diskutiert werden. Reeves und Nass vertreten den Standpunkt, dass Menschen ganz allgemein die Tendenz haben, Computer zu anthropomorphisieren. Die Interaktion mit Computern ruft bei Nutzern unweigerlich, eventuell nur unterbewusst, soziales Verhalten hervor.

Auch andere Autoren bestätigen, dass Computer von den Nutzern als soziale Akteure wahrgenommen und behandelt werden. D.h., Nutzer zeigen im Umgang mit Computern oft ähnliche Emotionen, Gedanken und Verhaltensweisen, wie gegenüber anderen Menschen. Für diese Sichtweise ist der Ausdruck »Social Response to Communication Technologies« mit dem Akronym »SCRT-Theorie« entstanden.

Bei Schnittstellen, die anthropomorphe Elemente (z.B. ein Gesicht oder eine Stimme) aufweisen, treten diese Effekte noch stärker auf. Beispielsweise empfinden Nutzer bei anthropomorphen Schnittstellen dem Computer gegenüber ein stärkeres Vertrauen und werden eher zu natürlichsprachlichen Eingaben angeregt. Allein die Präsenz eines menschlichen oder menschenähnlichen Gesichts bewirkt, dass Nutzer das System als unterhaltsamer bewerten und als intelligenter und glaubwürdiger einschätzen.

Ohne diese Effekte in Abrede stellen zu wollen, gehen die Interpretationen von Reeves und Nass einigen Autoren wie Sara Kiesler und Lee Sproull zu weit. Sie bezweifeln, dass Computer tatsächlich als soziale Akteure empfunden werden, und sind viel eher der Meinung, dass es sich bei den beobachteten Effekten nur um oberflächliche Phänomene handelt.

Selbst wenn Unsicherheit darüber besteht, ob Computer zurecht als soziale Akteure bezeichnet werden können, so sind die bisher beschriebenen Effekte durchweg positiv. Bei Untersuchungen mit anthropomorphen Figuren wurden jedoch auch negative Effekte beobachtet. Nutzer fühlen sich z.B. geradezu verärgert, wenn eine Schnittstelle menschenähnliche Qualitäten vermittelt, dann aber Regeln sozialen Verhaltens missachtet [Schaumburg 2001].

Die Präsenz anthropomorpher Figuren kann bei den Nutzern auch erhöhte Angst, gesteigerte Selbstüberwachung und mehr Fehler hervorrufen. Es wurde auch Impression Management beobachtet, d.h. die anthropomorphen Figuren stimulieren sozial erwünschtes Verhalten und Nutzer stellen sich selbst verstärkt positiv dar.

[Rosson/Carroll 2002] geben zu bedenken, dass natürlichsprachliche Interaktion nicht immer erstrebenswert ist. Denn auch wenn sie vieles vereinfacht, so könnte sie gegebenenfalls auch unpräziser und zeitaufwändiger sein als die Interaktion in einer künstlichen (Computer-)Sprache. Man sollte auch darauf achten, dass die Ansprache an die Nutzer nicht zu persönlich und freundlich ausfällt, was die Interaktion zwar anregen und unterhaltsamer machen könnte, die Nutzer aber ebenso irritieren und stören könnte.

Multimodalität kann an sich schon nachteilig sein, wenn man die Koordination der verschiedenen Kommunikationskanäle nicht ausreichend berücksichtigt. Es besteht auch die Gefahr der Reizüberflutung.

Anthropomorphe Schnittstellen und virtuelle Helfer werden oft als besonders nützlich für unerfahrene Nutzer angesehen. Erfahrene Nutzer hätten diese Unterstützungsform ja nicht mehr nötig. Doch besonders bei unerfahrenen Nutzern haben sich Probleme im Umgang mit einem virtuellen Helfer derart gezeigt, dass sich die Nutzer vom Helfer eher kontrolliert sahen als umgekehrt und sich

teilweise geradezu eingeschüchtert fühlten. Erfahrene Nutzer machten diese Erfahrungen nicht, sondern hatten viel mehr den Eindruck, die Kontrolle über den Helfer zu haben. Dies lässt virtuelle Helfer für späte Lernphasen wie beim kontinuierlichen Weiterlernen prinzipiell geeigneter erscheinen als für die anfängliche Einarbeitungszeit eines unerfahrenen Nutzers.

So wenig man auch bisher die Vorteile anthropomorpher Schnittstellen gegen ihre Nachteile abwägen kann, es ist bereits klar, dass die Wirkung einer Figur in erheblichem Maße von der jeweiligen spezifischen Ausgestaltung (Erscheinung, Sprache, nonverbales Verhalten) abhängt. Dabei scheint das jeweilige nonverbale Verhalten (Körperbewegungen, Gestik, Mimik) wichtiger zu sein als die bloße Präsenz oder das Aussehen.

Daher ist die prinzipielle Frage nach allgemeinen Verbesserungen durch den Einsatz anthropomorpher Schnittstellen auch viel weniger wichtig als die Frage nach dem Zusammenhang von Art und Ausprägung der anthropomorphen Schnittstelle, dem Kontext des Einsatzes und den beeinflussten Aspekten des Nutzungserlebnisses und der Performanz. Es gibt nur wenig Untersuchungen zur Effektivität oder Effizienz anthropomorpher Schnittstellen. Wie bereits erwähnt, ist auch nicht endgültig geklärt, in wie weit man von Computern im allgemeinen oder anthropomorphen Schnittstellen im speziellen tatsächlich von sozialen Akteuren sprechen kann. Welche Faktoren diese Sichtweise beeinflussen und andere Ansätze zur Erklärung der beobachteten Effekte wurden bisher zu wenig untersucht. Bei den bisherigen Untersuchungen handelt es sich auch fast nur um Laborstudien über kurze Zeiträume. Überprüfungen von Ergebnissen in Langzeituntersuchungen fehlen noch.

Eine Ausnahme bildet die bereits erwähnte Studie [Schaumburg 2001], deren Ergebnisse man direkt mit der entsprechenden Laborstudie [Microsoft 96] vergleichen kann. So kam die Laborstudie zu dem Ergebnis, dass der Microsoft Office-Assistent bei den Nutzern beliebter war als eine traditionelle Schnittstelle mit Dialog-Boxen und häufiger verwendet wurde. Dabei wurde er von den Nutzern subjektiv auch als effizienter wahrgenommen, obwohl objektiv das Gegenteil zutraf [Microsoft 96]. [Schaumburg 2001] lieferte jedoch als Ergebnis, dass der Office-Assistent bei Nutzern, die über längere Zeit mit ihm gearbeitet hatten, weniger beliebt war als die Schnittstelle mit Dialog-Boxen. Diese Langzeitstudie weist also gegenüber der Laboruntersuchung entgegengesetzte Ergebnisse auf, womit sie die Notwendigkeit für solche Untersuchungen unterstreicht.

Schließlich muss noch berücksichtigt werden, dass die heutigen technischen Möglichkeiten zur Implementierung anthropomorpher Schnittstellen gemessen an den hohen Ansprüchen relativ beschränkt sind. Verbales wie nonverbales Verhalten der bestehenden Systeme wirken noch zu künstlich. Mit dem technischen Fortschritt wird sich die Natürlichkeit der anthropomorphen Schnittstellen vergrößern, was eine erneute Überprüfung der Erkenntnisse erfordern wird.

Microsoft Bob

Microsoft Bob war ein Softwarepaket, das Anfang 1995 speziell für Computerneulinge, bei denen der Heimnutzen im Vordergrund steht, auf den Markt kam. Es war ein sehr früher Versuch, mit einer anthropomorphen Schnittstelle einen kommerziellen Erfolg zu erzielen, und sollte eine ganz neue Art von Software für eine neue Generation von Computernutzern sein. Dies stand nicht nur bei der Entwicklung der Anwendung selbst sondern auch bei der Verpackung, den Instruktionen-Materialien, der Werbekampagne und nicht zuletzt der Namensgebung im Vordergrund. Vor allem den Ängsten der Zielgruppe gegenüber Computern sollte begegnet werden, weswegen der Unterschied zu gewohnter Büro- und anderer Software besonders betont werden sollte. Anstatt des gewohnten Desktops sieht der Nutzer ein Zimmer als Zeichentrickbild [Winograd 96c].

Viele Objekte in dem Zimmer (siehe Abb. 3–9) sind reine Dekoration. Einige von ihnen sind animiert, andere dienen zum Start der Anwendungen von Bob. Durch Drücken der F1-Taste werden die Namen der funktionalen Objekte eingeblendet. Anstelle von Dialogboxen kommuniziert Bob mit dem Nutzer mittels animierter Charaktere (»Bobs Freunde«), aus denen man sich einen Führer durch die Anwendung aussuchen kann. Die verschiedenen Führer unterscheiden sich nicht nur äußerlich sondern auch in ihrer Ausdrucksweise voneinander [Lineback 2002]. Die Verwendung dieser Charaktere beruht auf Ergebnissen von Reeves und Nass (vgl. [Reeves/Nass 96]).

Um Bob möglichst einfach und überschaubar zu machen, beschränkte man sich auf einige wenige Standardanwendungen mit einfacher Funktionalität für die häufigsten Aufgaben. So verfügt Bob nicht über eine Textverarbeitung im eigentlichen Sinne. Da man nur aus wenigen Layouts für Briefe wählen

kann, wird die entsprechende Anwendung auch nur als Briefschreibe-Programm bezeichnet [Winograd 96c]. Die anderen integrierten Anwendungen sind ein Haushalts-Manager, ein privater Finanz-Manager, ein Adressbuch, ein Kalender, ein elektronisches Scheckbuch, das Geographiespiel GeoSafari und ein E-Mail-Programm.

Während der Arbeit werden oft akustische Signale ausgegeben: Beim Drücken von Knöpfen, bei der Auswahl animierter Objekte oder auch wenn der Führer nichts zu tun hat [Lineback 2002].



Abb. 3—9: Oberfläche von Microsoft Bob

Die Systemvoraussetzungen von einem 486-Prozessor mit mindestens 8MB Hauptspeicher und 32MB Platz auf der Festplatte waren für die Zeit sehr hoch [Arar o.J.]. Außer Hardware-Anforderungen wurde als Betriebssystem Microsoft Windows ab Version 3.1 vorausgesetzt. Tatsächlich war es damit für Computer-Neulinge eher ungeeignet. Es konnte nur von Leuten verwendet werden, die Windows 3.1 bereits besaßen und kannten. Den vorangegangenen Beschreibungen zufolge machten diese Nutzer dann wohl meist die Erfahrung, dass Bob selbst nur einen eingeschränkten Funktionsumfang gegenüber der gewohnten Windows-Version bietet, im Vergleich auch viel langsamer läuft – und das alles für einen Preis von 99 Dollar. Da kann es kaum verwundern, dass sich ein kommerzieller Erfolg nicht einstellte, die Fachpresse sich hauptsächlich lustig über ihn machte oder ihn ignorierte, und Microsoft Bob nach nur gut einem Jahr wieder vom Markt genommen wurde [Newman 99].

Es lassen sich noch weitere Gründe für Bobs Misserfolg anführen. Don Norman sagt, Bob verletze teilweise grundlegende softwareergonomische Regeln, ohne diese selbst näher aufzuzählen [Rheinfrank/Norman 96]. Wenn Bob sich aber beispielsweise dem Nutzer anpasst, indem er die am meisten verwendeten Programme in den Vordergrund schiebt, dann wirkt diese Anpassung einer Gewöhnung des Nutzers an die Anwendung entgegen. Weiterhin mag Norman darauf angespielt haben, dass es in Bob überhaupt keine standardmäßige Hilfe-Funktion gab. Statt dessen wurde man nur ständig gefragt, was man gerade machen möchte, damit Bob die passenden Optionen anzeigen kann. Wenn in diesen Optionen jedoch nicht die Lösung enthalten war, blieb ein bestehendes Problem

ungelöst. Es wurde nicht einmal ein Handbuch mitgeliefert. Die Dokumentation beschränkte sich auf eine Broschüre im Illustriertenstil mit Installationsanleitungen und Grundlagen zur Anwendung (vgl. [Newman 99], [Raskin 2000], [Arar o.J.]).

Genauer ist Normans Kritik nur bezüglich der animierten Charaktere. Die Ergebnisse von Reeves und Nass über die sozialen Interaktionen von Menschen mit unbelebten Objekten seien sehr interessant, wenn auch kontrovers. Man könne aus den Ergebnissen auch interessante Schlüsse ziehen, es sei jedoch ein sehr großer Schritt, zu folgern, dass man ein Gerät mit einer Persönlichkeit ausstatten sollte, nur weil Menschen Geräten im Umgang mit ihnen Persönlichkeiten verleihen. Dies könnte sich nicht auf die Forschungsergebnisse stützen. Überhaupt hätte es keinerlei Ergebnisse gegeben, aus denen man überhaupt etwas für die Konstruktion von Geräten hätte folgern können. Die Entwickler von Microsoft Bob hätten also einige Ergebnisse von Reeves und Nass überinterpretiert, indem sie versucht haben, besonders niedliche Charaktere zu entwickeln [Rheinfrank/Norman 96]. Solch eine simulierte ›Freundlichkeit‹ kann einer Interaktion zwar mehr Liebenswürdigkeit und Spaß verleihen, aber gleichzeitig Neulinge in die Irre führen und erfahrene Nutzer verärgern [Rosson/Carroll 2002].

4. Ein neues Konzept zur Lernunterstützung: Zeitverzögerte Nutzungshinweise

»Lotse ein erfahrener Schiffsoffizier, der sein See-, Fluss- oder Kanalrevier so genau kennt, dass er den Schiffskapitän zuverlässig beraten kann. Dieser trägt jedoch auch in Revieren mit Lotsenzwang stets alle Verantwortung.«
Erklärung des Begriffes Lotse auf <http://www.wissen.de> (16.06.2003)

In diesem Kapitel stelle ich zwei Ansätze zur Erleichterung kontinuierlichen Weiterlernens vor, die ich aus den vorangegangenen Betrachtungen abgeleitet habe. Diese Ansätze wurden für die Online-Brokerage-Anwendung *maxblue* implementiert. Ich werde die Ergebnisse dieses Praxistests schildern und daran anschließend eine Bewertung der entwickelten Lösungen vornehmen.

4.1. Konzept

Nutzungshinweise können nur dann effektiv sein, wenn ihr Adressat bei ihrem Erscheinen dafür auch empfänglich ist. Ausgehend von der Aussage, dass Erklärungen den Nutzern mehr helfen, wenn sie nach Vollendung einer Tätigkeit gegeben werden (siehe Abschnitt 2.5.1 Minimalismus, S. 14f), entstand die Idee, Nutzungshinweise verzögert gegenüber der Aufgabenerledigung auszugeben. So soll eine Störung des Nutzers durch das Erscheinen des Hinweises vermieden werden. Er wird bei seiner aktuellen Arbeit nicht unterbrochen, sondern erhält den Hinweis erst, nachdem er seine Aufgabe erledigt hat. Dies hat nicht nur den Vorteil, dass die Aufnahmefähigkeit wahrscheinlich größer sein wird als in der vorangegangenen Konzentrationsphase. Zusätzlich vorteilhaft ist, dass der Nutzer in der Anwendungssituation Erfahrungen gesammelt hat, worauf in den verzögerten Hinweisen Bezug genommen werden kann.

Verzögerte Hinweise erscheinen im Kontext des normalen, alltäglichen Umgangs mit einer Anwendung, wie es für kontinuierliches Weiterlernen gefordert wird. Für die Einarbeitung in eine noch unbekanntere Anwendung sind verzögerte Hinweise weniger geeignet, da die Gefahr besteht, dass die Hinweise für einen unerfahrenen Nutzer zu spät ausgegeben werden.

In der Klassifikation der Lernkomponenten, Hilfe- und Assistenzsysteme (vgl. Abschnitt 3.1 Klassifikation, S. 19ff) sieht man, dass es bisher kaum Ansätze gibt, in denen die Ausgabe verzögert stattfindet. Die Übersicht enthält nur zwei solcher Ansätze. Die Nutzer- und Expertenmodellierung ist

jedoch eher als Forschungsprojekt zu bezeichnen, und Newsgroups und FAQs sind selbst nicht Bestandteile von Anwendungen.

Um allgemeine Aussagen über die Eignung von verzögerten Nutzungshinweisen zu bekommen, wird das Maß der Zeitverzögerung zwischen zwei Ansätzen variiert. In beiden Fällen beruhen die gegebenen Hinweise auf Analysen von Nutzungshistorien. Die Verzögerung bedingt es, dass die Hinweise nur beschreibend sind, da für Verweise in die Anwendung oder eine aktive Unterstützung zum Ausgabezeitpunkt der nötige Anwendungskontext nicht mehr existiert.

4.1.1. Verzögerte Explorationskarten

Bei verzögerten Explorationskarten erfolgt die Ausgabe noch während der Nutzer mit der Anwendung arbeitet. Der Nutzer bekommt die Hinweise in einem separaten Fenster jeweils kurz nach der Erledigung einer Aufgabe.

Die Hinweise werden in Form von Explorationskarten gegeben, weil sie eine wirkungsvolle Umsetzung des minimalistischen Erklärungsansatzes sind und besonders viel versprechend für kontinuierliches Weiterlernen erscheinen.

Der wesentliche Unterschied zu den Explorationskarten, wie sie im Kapitel über Explorationsunterstützung (Abschnitt 3.7.4, S. 35) beschrieben werden, ist, dass verzögerte Explorationskarten nicht auf Anforderung eines Nutzers ausgegeben werden, indem er aus einer Liste vorbereiteter Explorationskarten auswählt. Sie werden vielmehr zu einem möglichst passenden Zeitpunkt automatisch angezeigt. Damit sollen die Vorteile von Explorationskarten, die den minimalistischen Lernansatz durch kurze, unvollständige, aufgabenorientierte Erklärungen umsetzen und den Nutzer zur weiteren Arbeit mit dem System anregen, für eine automatische Lernempfehlung genutzt werden.

4.1.2. Verzögerte intelligente Hilfe

Bei verzögerter intelligenter Hilfe werden die Hinweise in E-Mails an die Nutzer verschickt. Die Ausgabe erfolgt dabei mit einem großen zeitlichen Abstand gegenüber der Aufgabenerledigung. Die Distanz zur Anwendung wird zusätzlich dadurch vergrößert, dass das Medium der E-Mails außerhalb der Anwendung liegt. Diese Form soll besonders unaufdringlich sein und die Steuerung durch den Nutzer gewährleisten, da er selbst bestimmen kann, ob und wann er den Hinweis in einer E-Mail lesen möchte.

Auch bei dieser Ausgabeform von Nutzungshinweisen sollten die beschriebenen Prinzipien des Minimalismus (Abschnitt 2.5.1, S. 14f) beachtet werden. Bei E-Mails ist es darüber hinaus sehr wichtig, dass dem Nutzer der Zusammenhang von Hinweis und Anwendung deutlich vermittelt wird. Dazu müssen bereits die Betreffzeile und sogar der Absender der E-Mail so gewählt werden, dass der Nutzer die Verbindung herstellen kann. Es wäre ideal, wenn der Nutzer dadurch sein Vorwissen bereits aktiviert hat, wenn er beginnt, den Text der E-Mail zu lesen.

Da Nutzer es noch nicht gewohnt sind, dass sie persönliche E-Mails in dieser Art von einer Anwendung zugeschickt bekommen, kann man damit rechnen, dass sie auf den Inhalt der E-Mails neugierig sind. Damit kann ihre intrinsische Motivation erhöht und die Grundlage für einen großen Lernerfolg geschaffen werden.

4.2. Prototypische Implementation: Der *maxblue* Lotse

Die Firma SinnerSchrader ermöglichte es mir, die Ideen des vorgestellten Konzeptes in *maxblue* (<http://www.maxblue.de>), dem Onlinebroker der Deutschen Bank, praktisch umzusetzen. Die Web-Anwendung *maxblue* bietet vielfältige Informationsmöglichkeiten für Wertpapierbesitzer. Den Nutzern stehen Marktinformationen der Deutschen Bank sowie Daten und Nachrichten internationaler

Informationsdienste (Risk Grades, Reuters, Financial Times, Innovative Software/Teledata, VWD etc.) zur Verfügung. Für Kunden der Deutschen Bank ist auch das Handeln mit Wertpapieren möglich.



Abb. 4—1: Homepage von *maxblue*

Im Kontext dieser Diplomarbeit sind besonders die Funktionen interessant, die sich nicht speziell auf die Wertpapierthematik beziehen, sondern die auch in anderen Kontexten einsetzbar sind. Dazu zählen u.a. anwendungsspezifische Lesezeichen, die registrierte Nutzer für die Seiten anlegen können, die sie immer wieder aufrufen möchten. Da diese Funktionalität von relativ wenigen Nutzern verwendet wird, wurde sie in Abstimmung mit der Deutschen Bank als Inhalt gewählt, der durch die beiden Ansätze vermittelt werden soll. Um zu verdeutlichen, dass den Nutzern interessante Hinweise für eine effizientere Nutzung von *maxblue* zur Verfügung stehen, wurden beide Ansätze unter dem Begriff »*maxblue* Lotse« zusammengefasst. Die Metapher eines Lotsen vermittelt, dass der Nutzer Ratschläge eines kompetenten Experten erwarten kann, dabei in seiner Kontrolle über die Anwendung aber nicht eingeschränkt wird.

Die möglichen Effizienzsteigerungen, die sich durch die Verwendung der *maxblue* Lesezeichen für die Nutzer ergeben, sollen anhand von fünf Beispielen verdeutlicht werden.

- [1] »Akte Detailübersicht«: Die Detailübersicht zu einer Aktie kann man über das Navigationsmenü in 4 Schritten (Märkte – Aktien – einen Index auswählen – eine Aktie auswählen) oder über die Kursabfrage aufrufen. Wenn sich ein Nutzer immer wieder für dieselbe Aktie interessiert, kann ein Lesezeichen diesen Weg auf einen Klick abkürzen. Selbst mit eindeutiger Suche über die Kursabfrage ergeben sich Vorteile, da mit Lesezeichen keine Wechsel mehr zwischen Maus und Tastatur für die Texteingabe nötig sind.

Ein besonders positiver Effekt ergibt sich für Nutzer, die sich für einen speziellen Börsenplatz interessieren. Der für diese Auswahl nötige Extraklick kann durch Anlegen eines entsprechenden Lesezeichens ebenso eingespart werden.

- [2] »Aktie Analysten«: Der Weg zum Aufrufen von Analystenmeinungen zu einer Aktie verläuft genauso wie der zur »Detailübersicht« und erfordert danach einen weiteren Klick. Über das Navigationsmenü benötigt man also 5 Klicks oder zusätzlich nach der Kursabfrage einen Klick. Durch Verwendung eines Lesezeichens ist hier die Einsparung also noch größer.
- [3] »Fonds Top/Flop«: Die Liste mit den Tops und Flops der Fonds erreicht man über die Navigation in 3 Schritten (Märkte – Fonds – Top/Flop Markt). Wenn man an einem anderen als dem Standardzeitraum von einem Jahr interessiert ist, ist ein weiterer Klick notwendig. Mit einem Lesezeichen können diese 3 oder 4 Schritte auf einen verkürzt werden.
- [4] »Börsenkalender«: Der Börsenkalender ist über das Navigationsmenü 3 Schritte (Märkte – News – Börsenkalender) von der Homepage entfernt. Durch die Verwendung eines Lesezeichens für den Börsenkalender kann diese Distanz auf einen einzigen Schritt reduziert werden.
- [5] »Virtuelles Depot Übersicht Statistik«: Wenn sich ein Nutzer eines seiner Virtuellen Depots ansehen möchte, benötigt er dafür über die Navigationsleisten 3 Schritte (Märkte – Virtuelles Depot – Depotname). Mit einem Lesezeichen benötigt der Nutzer dafür wiederum nur einen einzigen Schritt. Da nur angemeldete Nutzer ein Virtuelles Depot angezeigt bekommen, ergibt sich in diesem Fall auch kein Zusatzaufwand dadurch, dass man für das Aufrufen der Lesezeichen vorher die Anmeldung durchgeführt haben muss.



Abb. 4–2: Ausschnitt des *maxblue* Navigationsmenüs mit Lesezeichen

Durch die Verwendung von Frames und JavaScript in der *maxblue* Anwendung sind diese Effekte mit normalen Browser-Lesezeichen nicht zu realisieren. Außerdem bieten die speziellen *maxblue* Lesezeichen den Vorteil, dass sie für den Nutzer durch die Integration in das obere Navigationsmenü jederzeit sichtbar sind. Sie können somit schneller aufgerufen werden, als dies bei Browser-Lesezeichen möglich ist, für die im Allgemeinen erst ein Menü geöffnet werden muss. In Abbildung 4–2 ist das 3-zeilige Navigationsmenü von *maxblue* zu sehen. Wenn sich ein Nutzer angemeldet hat, erscheinen in der Zeile darüber der Link »My maxblue« auf die Seite zur Verwaltung der persönlichen Daten und Einstellungen, der Link »Logout« zum Abmelden und daneben die Liste der angelegten Lesezeichen.

4.2.1. Explorationskarten in Popup-Fenstern

Mit den verzögerten Explorationskarten soll den Nutzern allgemein der Gebrauch der *maxblue* Lesezeichen erklärt werden.

maxblue Lotse: Lesezeichen

Wussten Sie schon, dass ...?

... Sie mit der Lesezeichen-Funktion innerhalb von My maxblue direkten Zugriff auf von Ihnen gewählte Services haben? Es geht ganz einfach:

1. Sie rufen die Lesezeichen-Funktion mit Klick auf das Icon am rechten oberen Seitenrand auf.

Deutsche Bank

kläufigen Umsatz Intraday DAX

2. Anschließend können Sie dem Lesezeichen einen von Ihnen gewählten Namen zuweisen.

Lesezeichen setzen - Microsoft Internet ...

Lesezeichen

Virtuel 1

- ▶ Bitte geben Sie diesem Lesezeichen einen Namen mit maximal 10 Zeichen und speichern Sie es.

Mein Depot **Speichern**

3. Nach Klick auf „Speichern“ erscheint das Lesezeichen in der Leiste über dem maxblue Logo.

My maxblue | LogOut Mein Depot

Home | Märkte | Analyse | Tra

maxblue

Unter My maxblue/Lesezeichen können Sie jederzeit Ihre Lesezeichen umbenennen, neu sortieren oder löschen.

Ich möchte keine weiteren Hinweise zur Bedienung von My maxblue als PopUp-Fenster erhalten. Unter My maxblue/Ihre Daten kann ich diese Funktion jederzeit wieder aktivieren.

Fenster schließen

Abb. 4—3: Explorationskarte für die Lesezeichen von *maxblue*

Wie bei Explorationskarten üblich steht am Anfang eine Beschreibung der Aufgabe, damit die Nutzer sofort sehen, ob sie den Hinweis verwenden möchten. Dann folgt die eigentliche Erklärung zur Lösung der Aufgabe in Form einer illustrierten Darstellung, in der das Anlegen eines Lesezeichens beschrieben wird. Das unterste Bild dient der Erfolgskontrolle. Den Nutzern wird gezeigt, wo angelegte Lesezeichen hinterher erscheinen. Da die Aufgabe nicht komplex ist, wurde auf eine Erläuterung häufiger Fehler verzichtet. Unten auf der Seite befindet sich eine Checkbox, mit der das weitere Erscheinen von Explorationskarten unterbunden werden kann. Somit hat der Nutzer bei jedem Erscheinen einer Explorationskarte die volle Kontrolle über diese Ausgabeform des Lotsen.

Per Web Mining wurden geeignete Plätze zum Anzeigen der Explorationskarten identifiziert. Aus den Zugriffszahlen wurde ermittelt, welche Seiten von *maxblue* durch die Nutzer am meisten aufgerufen werden. Für diese hoch frequentierten Seiten wurden anschließend Clickstream-Analysen durchgeführt, um zu sehen, für welche Seiten besonders lange Navigationspfade verwendet werden.

Dahinter stand die Annahme, dass die Nutzer bei diesen Seiten am meisten von der Verwendung von Lesezeichen profitierten. Auf diese Weise wurden 8 Seiten identifiziert, die von den Nutzern sehr oft, aber meist erst nach vielen Zwischenschritten aufgerufen werden. Die Explorationskarten wurden so in die Anwendung eingebaut, dass sie in einem Popup-Fenster angezeigt wurden, wenn ein Nutzer eine der 8 Seiten verlies. Diese Ausgabeform wurde auf den *maxblue*-Seiten den Nutzern als »Popup-Option« des Lotsen angeboten.

Für das Ausschalten der Popup-Fenster war es nicht nötig, dass ein Nutzer sich vorher angemeldet hatte, da für diese Einstellung persistente Cookies verwendet wurden. Damit die Nutzer die Anzeige von Popup-Fenstern wieder einschalten konnten, gab es eine weitere Checkbox auf der Seite zur Verwaltung der persönlichen Daten im »My maxblue«-Bereich.

Bei *maxblue* besteht nicht die Gefahr, dass Popup-Fenster mit Werbeeinblendungen verwechselt werden. Popup-Fenster werden bereits an mehreren Stellen von *maxblue* eingesetzt, so dass die Nutzer an sie gewöhnt sind und wissen, dass sie keine Werbung erwarten müssen.

4.2.2. Verzögerte Hilfe per E-Mails

Bei der verzögerten intelligenten Hilfe wurde in den E-Mails ebenfalls allgemein der Gebrauch der *maxblue* Lesezeichen erklärt. Zusätzlich gab es jedoch auch einen individuellen Teil, in dem die Nutzer darauf hingewiesen wurden, wie sie persönlich am meisten von Lesezeichen profitieren können. Der vollständige Text einer E-Mail und das jeweils angehängte PDF-Dokument, das der Illustration der rein textuellen Erklärung in der E-Mail dienen sollte, sind im Anhang abgedruckt (Abschnitt 6.2 Beispiel einer vom Lotsen versendeten E-Mail, S. 69f).

Den Nutzern wurde diese Möglichkeit als »E-Mail-Option« des Lotsen dargestellt. Wenn sie diese Option einschalteten, wurden sämtliche ihrer Funktionsaufrufe individuell, nicht anonymisiert protokolliert. Dies diente dazu, den Nutzern den speziellen, persönlichen Hinweis zu geben, wo sie den größten Gewinn durch Lesezeichen erzielen können. Falls diese Option ausgeschaltet wurde, so fand keine individualisierte Protokollierung statt.

Wie die Nutzer diese Option ein- und wieder ausschalten konnten, ist anhand der entsprechenden Screenshots dargestellt, die sich im Anhang befinden (Abschnitt 6.1 Screenshots des *maxblue* Lotsen, S. 62ff).

Der Zusammenhang einer E-Mail mit *maxblue* wurden den Nutzern dadurch vermittelt, dass der Absender der E-Mails »lotse@maxblue.de« lautete und bereits in der Betreffzeile das Thema der Lesezeichen von *maxblue* angesprochen wurde.

4.3. Evaluation des Lotsen

Der Lotse war bei *maxblue* vom 26.03. bis 08.05.2003 online im Einsatz. Im Folgenden wird ein Vorher-Nachher-Vergleich bezüglich des Gebrauchs von Lesezeichen durch die Nutzer von *maxblue* durchgeführt. In diesem Abschnitt werden die Auswertungsergebnisse vorgestellt, um im nächsten Abschnitt bewertet zu werden.

Für den Vorher-Zeitraum wurden Daten vom 06.03. bis 19.03.2003 verwendet. Der Nachher-Zeitraum erstreckte sich vom 26.03. – dem Tag der Onlinestellung des Lotsen – über 4 Wochen bis zum 24.04.2003. Die statistischen Analysen wurden mit dem Programm SAS (Statistical Analysis System) in der Version 8.2 der Firma SAS Institute Inc. vorgenommen. Mit diesem Programm wurde für die verfügbaren Daten ein Duncan Grouping durchgeführt. Die Werte von p geben jeweils die Wahrscheinlichkeit an, dass die Änderung der Werte vom Vorher-Zeitraum zum Nachher-Zeitraum zufällig erfolgte. Ich werte ein Ergebnis von $p < 0,01$ als statistisch signifikante Änderung zwischen den beiden Zeiträumen.

Zu Vergleichszwecken wurde ein Durchschnittswert ermittelt, der angibt, wie viele Lesezeichen zwischen Oktober 2002 und Februar 2003 durchschnittlich an einem Tag neu angelegt wurden. Dieser

Durchschnittswert diente zur Extrapolierung, um zu sehen, wie sich die Anzahl der Lesezeichen ohne den Einfluss des Lotsen entwickelt hätte.

Wie haben sich die Nutzerzahlen und die Zahl der Lesezeichen entwickelt?

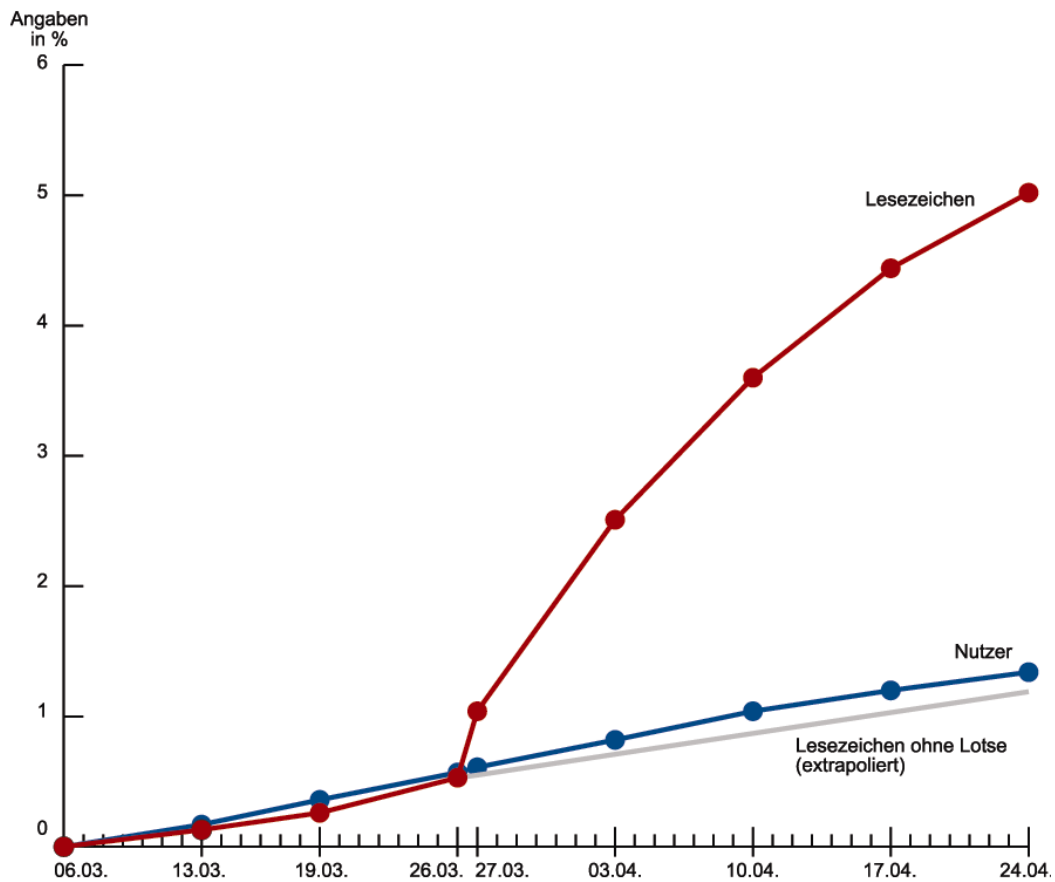


Abb. 4—4: Entwicklung der Nutzerzahlen und der Anzahl an Lesezeichen

	März 2003					April 2003				Mittelwert		p
	06.	13.	19.	26.	27.	03.	10.	17.	24.	vorher	nachher	
Nutzer	0,00	0,17	0,36	0,57	0,61	0,82	1,04	1,20	1,34	0,18	0,93	0,0408
Lesezeichen ohne Lotse (extrapoliert)	0,00	0,13	0,26	0,53	0,55	0,71	0,87	1,03	1,19	0,13	0,81	0,0687
Lesezeichen mit Lotse	0,00	0,13	0,26	0,53	1,04	2,51	3,60	4,44	5,02	0,13	2,86	0,0045

Tab. 4—5: Entwicklung der Nutzerzahlen und der Anzahl an Lesezeichen

Angegeben ist jeweils der prozentuale Anstieg des entsprechenden Wertes. Im Gegensatz zur Anzahl der bei *maxblue* registrierten Nutzer und der extrapolierten Anzahl von Lesezeichen ist die Anzahl der tatsächlich angelegten Lesezeichen statistisch signifikant gestiegen. Die höhere Steigerung der Anzahl von Lesezeichen ist eindeutig auf den Einsatz des Lotsen zurückzuführen.

Wie viele Nutzer haben Lesezeichen angelegt?

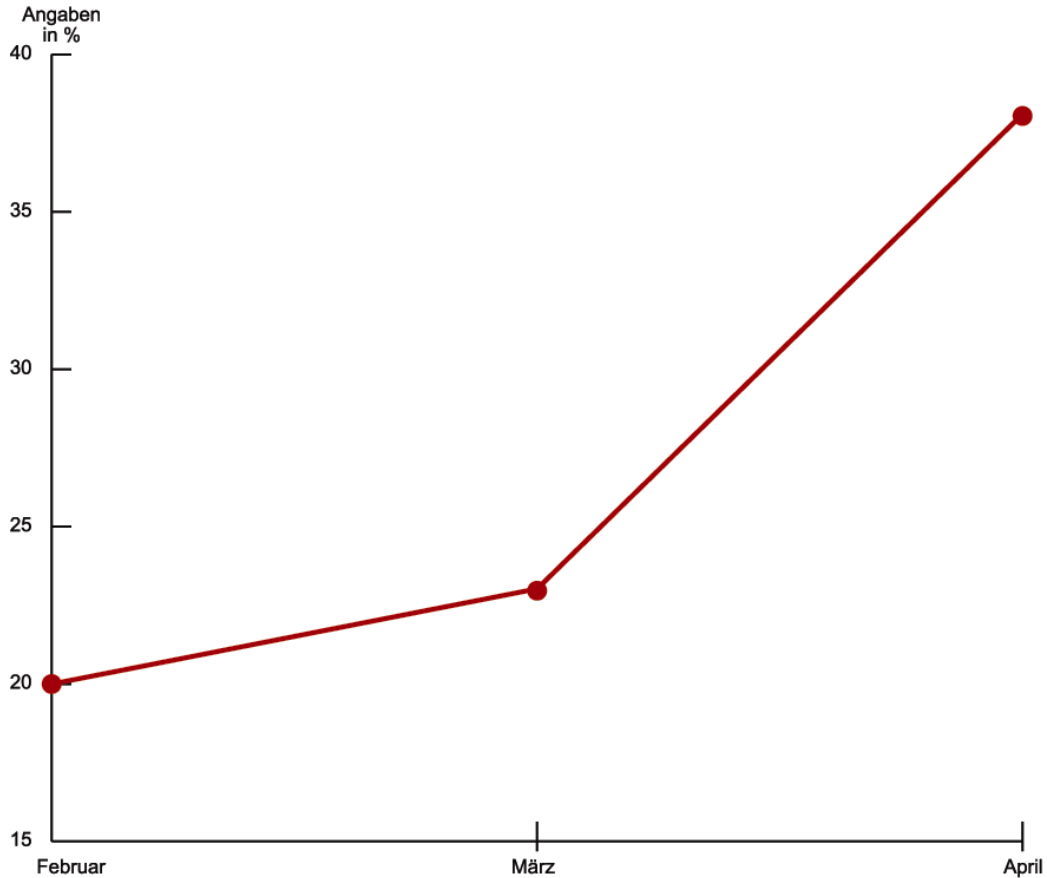


Abb. 4—6: Anteil der aktiven Nutzer mit Lesezeichen

Februar	März	April	Mittelwert		p
			vorher	nachher	
19,96%	21,05%	36,13%	20,51	36,13	0,158

Tab. 4—7: Anteil der aktiven Nutzer mit Lesezeichen

Bei den aktiven Nutzern wurden nicht alle registrierten Nutzer berücksichtigt, sondern nur solche, die seit Januar 2003 auch tatsächlich *maxblue* verwendet haben.

Der März wurde nur bis zum 25.3. gewertet, bevor der Lotse online ging. Die Märztagte ab dem 26.3. wurden für die Berechnung dem April zugeschlagen, um einen sauberen Vorher-Nachher-Vergleich zu bekommen.

Aufgrund einer ungenügenden Datenerfassung in der Zeit, als der Lotse online war, mussten diese Daten nachträglich ermittelt und aufbereitet werden. Eine exakte Datenerfassung für jeden Tag war im Nachhinein nicht mehr möglich. Aus diesem Grund standen für die statistische Auswertung nur sehr wenig Werte zur Verfügung, so dass die Aussage, der Anteil an aktiven Nutzern mit Lesezeichen sei nicht statistisch signifikant gestiegen, nur unter Vorbehalt vorgenommen werden kann.

Wie viele Lesezeichen werden täglich neu angelegt?

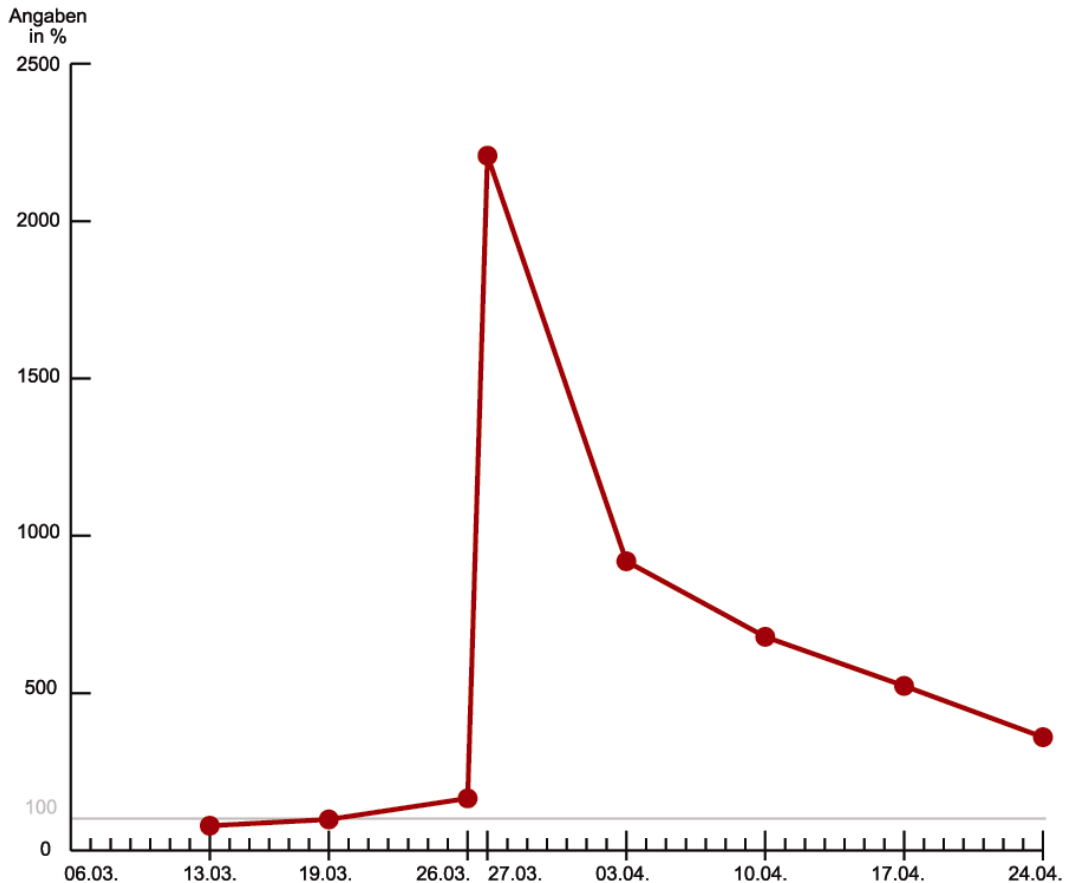


Abb. 4—8: Täglicher Zuwachs an Lesezeichen

06. bis 12.03.	13. bis 19.03.	20. bis 26.03.	nur am 27.03.	28. bis 03.04.	04. bis 10.04.	11. bis 17.04.	18. bis 24.04.	Mittelwert vorher nachher		p
78,58	98,58	165,50	2208,00	919,00	679,00	523,00	360,00	88,60	809,10	0,0092

Tab. 4—9: Täglicher Zuwachs an Lesezeichen

Angegeben sind Prozentwerte für die aufgeführten Zeiträume. 100% entsprechen dem Durchschnittswert der täglich angelegten Lesezeichen, der für den Zeitraum Oktober 2002 bis Februar 2003 berechnet wurde. Der tägliche Zuwachs der Lesezeichen ist statistisch signifikant gestiegen.

Welche Wirkung hat der Lotse?

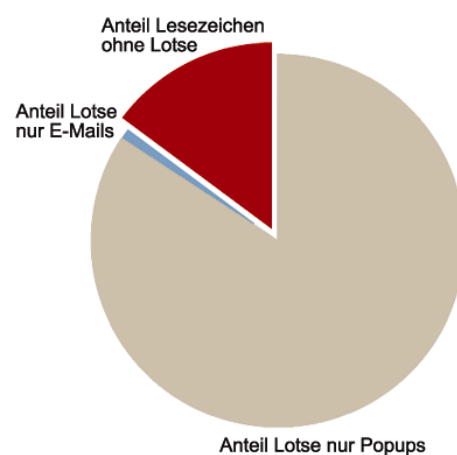


Abb. 4—10: Wirkung von Popups und E-Mails

Anteil Lotse nur Popups	84,24%
Anteil Lotse nur E-Mails	0,94%
Lesezeichen ohne Lotse (extrapoliert)	14,82%

Tab. 4—11: Wirkung von Popups und E-Mails

85,12% der im Beobachtungszeitraum angelegten Lesezeichen sind auf den Einfluss des Lotsen zurückzuführen (Summe aus Popups und E-Mails). Fast die gesamte Wirkung ist der Ausgabeform der Popups zuzuschreiben. Lediglich unter 1% der angelegten Lesezeichen beruht auf den vom Lotsen versandten E-Mails.

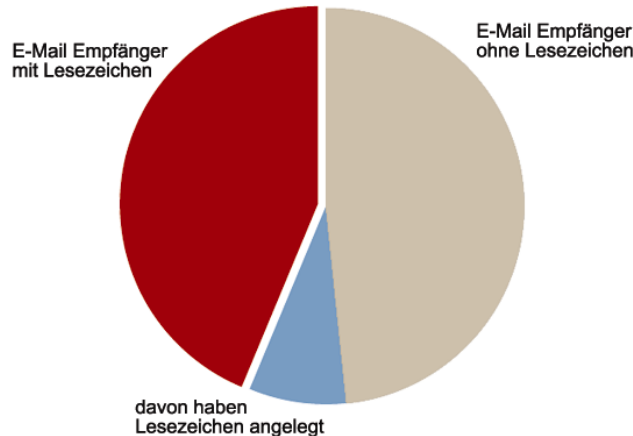


Abb. 4—12: Profil der E-Mail-Empfänger

E-Mail-Empfänger ohne Lesezeichen	56,29%
davon haben Lesezeichen angelegt	7,95%
E-Mail-Empfänger mit Lesezeichen	43,71%

Tab. 4—13: Profil der E-Mail-Empfänger

43,71% der Nutzer, die sich vom Lotsen eine E-Mail haben schicken lassen, hatten bereits Lesezeichen angelegt. Das ist ein höherer Anteil an Lesezeichen als bei den übrigen aktiven Nutzern, von denen im April 36,13% Lesezeichen angelegt hatten. Über die Gründe dafür könnte nur spekuliert werden, aber es ist interessant, dass die E-Mail-Empfänger offenbar eine besondere Gruppe von *maxblue*-Nutzern darstellen.

Keine Lesezeichen angewendet haben vor Empfang der E-Mail des Lotsen 56,29% der Nutzer. Von diesen Nutzern wurden 14,12% durch die E-Mail angeregt, hinterher Lesezeichen anzulegen. Umgerechnet auf alle E-Mail-Empfänger ergibt sich der Anteil von 7,95%.

Wie werden die Lesezeichen tatsächlich verwendet?

Es reicht nicht aus, sich nur die Anzahl und Entwicklung der angelegten Lesezeichen anzusehen. Man muss auch die tatsächliche Verwendung der Lesezeichen auf den verschiedenen Seiten von *maxblue* verfolgen. Die wichtigsten charakteristischen Ergebnisse dazu sind hier in graphisch aufbereiteter Form dargestellt.

Bei 4 Seiten gibt es einen statistisch signifikanten Anstieg der Seitenaufrufe über Lesezeichen. Bei den anderen Seiten ist der Anteil, der über Lesezeichen aufgerufen wird, meist sehr gering. Viele dieser Seiten befinden sich im so genannten »öffentlichen« Bereich von *maxblue*, der jedermann auch ohne Anmeldung bei *maxblue* zugänglich ist. Diese Seiten werden oft von Nutzern aufgerufen, die nicht registriert sind und daher auch gar keine Lesezeichen anlegen können. Deutliche Ergebnisse können dort nicht mehr beobachtet werden. Das Virtuelle Depot gehört hingegen zum »geschlossenen« Bereich, der nur für registrierte Nutzer nach einer Anmeldung zugänglich ist. Von den 4 Seiten, für die ein statistisch signifikanter Anstieg beobachtet wurde, gehören 2 zu diesem Virtuellen Depot im

geschlossenen Bereich. Dies sind die Seiten »VD Übersicht Statistik« und »Watchlist«, die im Virtuellen Depot auch am häufigsten aufgerufen werden.

Bei der Seite »Übersicht Index-Zertifikat« gibt es sogar einen statistisch signifikanten Abstieg der Aufrufe über Lesezeichen. Dies ist mit darauf zurückzuführen, dass diese Seite zu den am wenigsten aufgerufenen Seiten gehört, die hier betrachtet werden. Bei den geringen Aufrufzahlen der Seite sind die Schwankungen im Nutzungsverhalten sehr groß. Außerdem standen für die Seitenaufrufe im Vorher-Zeitraum nur Daten vom 06.03. bis zum 13.03. zur Verfügung. Große Schwankungen bei den wenig aufgerufenen Seiten in diesem relativ kurzen Vorher-Zeitraum haben das Erkennen weiterer statistisch signifikanter Anstiege erschwert und in diesem Fall sogar einen statistisch signifikanten Abstieg ergeben.

Auf den folgenden Graphiken ist visualisiert, wie viel Prozent der Aufrufe der angegebenen Seiten über ein Lesezeichen stattfanden. Die durchgezogenen Linien geben für den jeweiligen Tag im Vorher- und im Nachher-Zeitraum den genauen Prozentwert an. Vorher- und Nachher-Zeitraum sind nur durch gepunktete Linien verbunden, da für die Zwischenzeit keine Daten zur Verfügung stehen. Für jede Kurve ist eine Fläche eingezeichnet, deren Höhe den statistisch berechneten Mittelwert an Lesezeichenaufrufen für diese Seite im entsprechenden Zeitraum angibt. Die Oberkante einer Fläche gibt also jeweils den Mittelwert der farblich zugehörigen Kurve wieder. Die Darstellung der Fläche wurde gewählt, um die Graphiken nicht durch eine zusätzliche dünne Linie unübersichtlich zu gestalten. So bleibt trotz der Informationsfülle der jeweilige Mittelwert erkennbar und man sieht einerseits, wie sehr die Kurven um die Mittelwerte schwanken, und andererseits, wie sehr sich der Mittelwert des Nachher-Zeitraumes gegenüber dem des Vorher-Zeitraumes verändert hat. Bei vielen Seiten sieht man deutlich, dass am Wochenende prozentual viel weniger Seitenaufrufe über Lesezeichen stattfanden als während der Arbeitstage. Aufgrund dieser Schwankungen wären die Veränderungen bei der Nutzung der Lesezeichen ohne Visualisierung der Mittelwerte in den Graphiken kaum erkennbar.

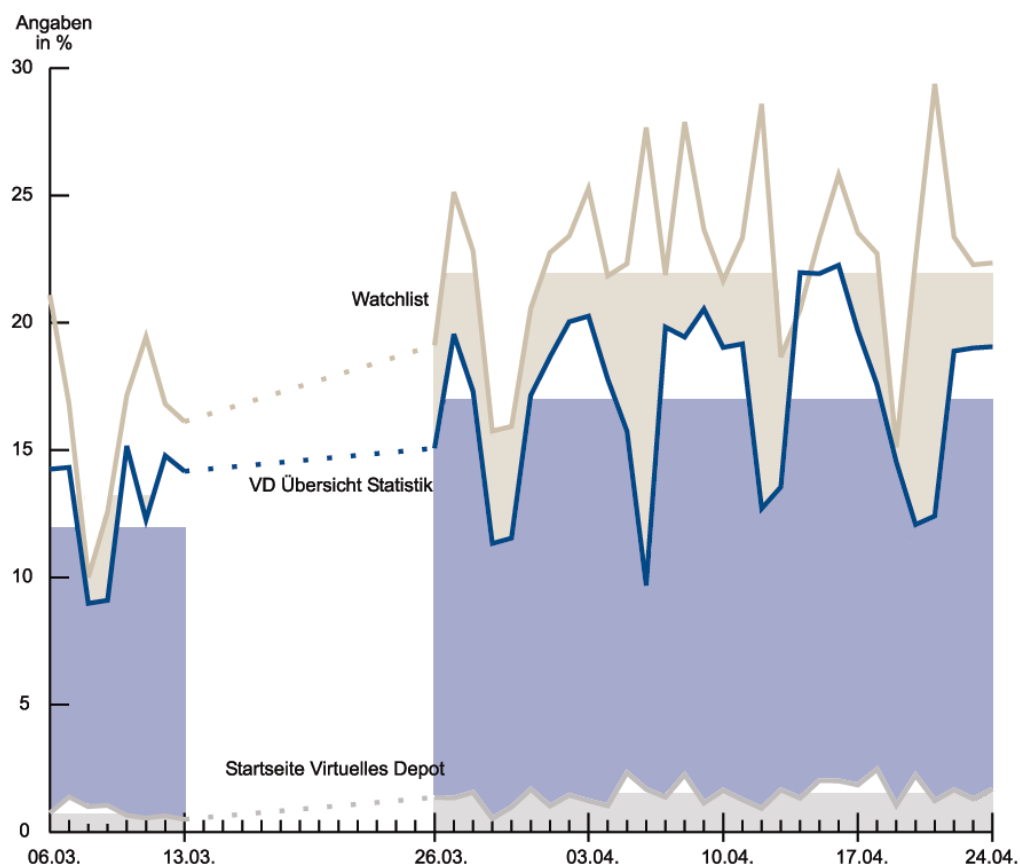


Abb. 4–14: Seitenaufrufe über Lesezeichen im Virtuellen Depot

Für »VD Übersicht Statistik« und »Watchlist« ist der Anstieg statistisch signifikant.

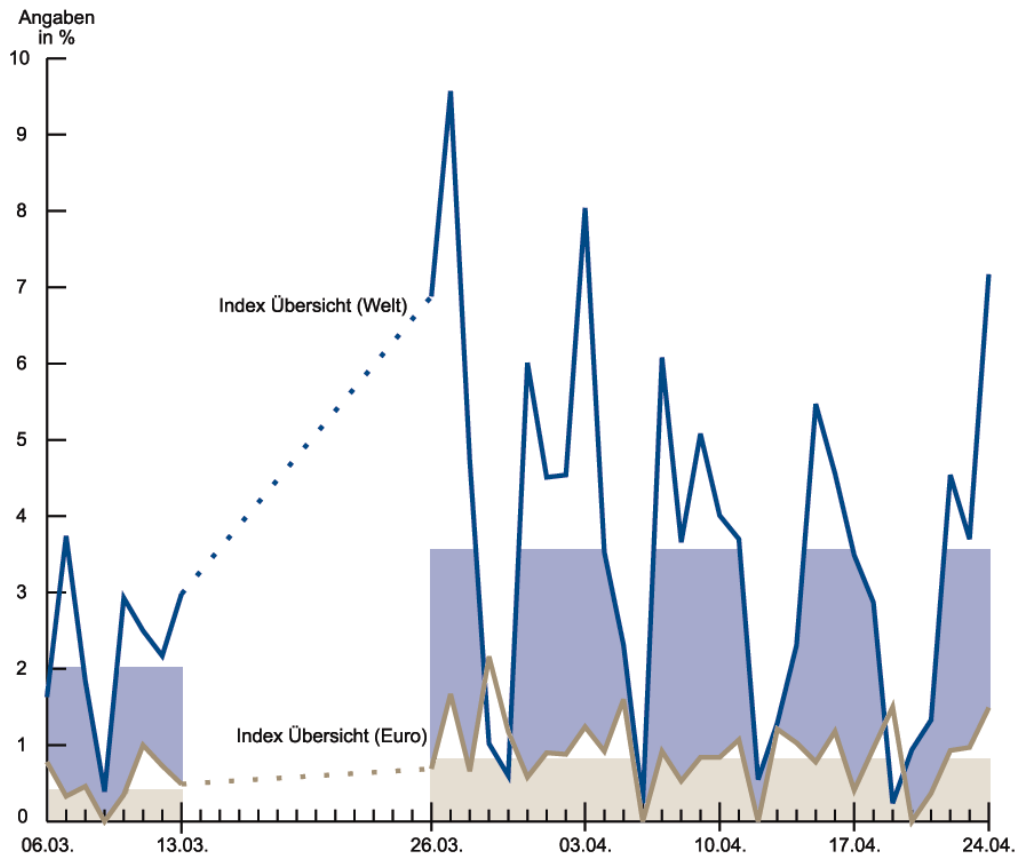


Abb. 4–15: Seitenaufrufe über Lesezeichen bei den Index-Übersichten

Für »Index Übersicht (Euro)« (Abb. 4–15) und »Aktie Chart« (Abb. 4–16) ist der Anstieg statistisch signifikant.

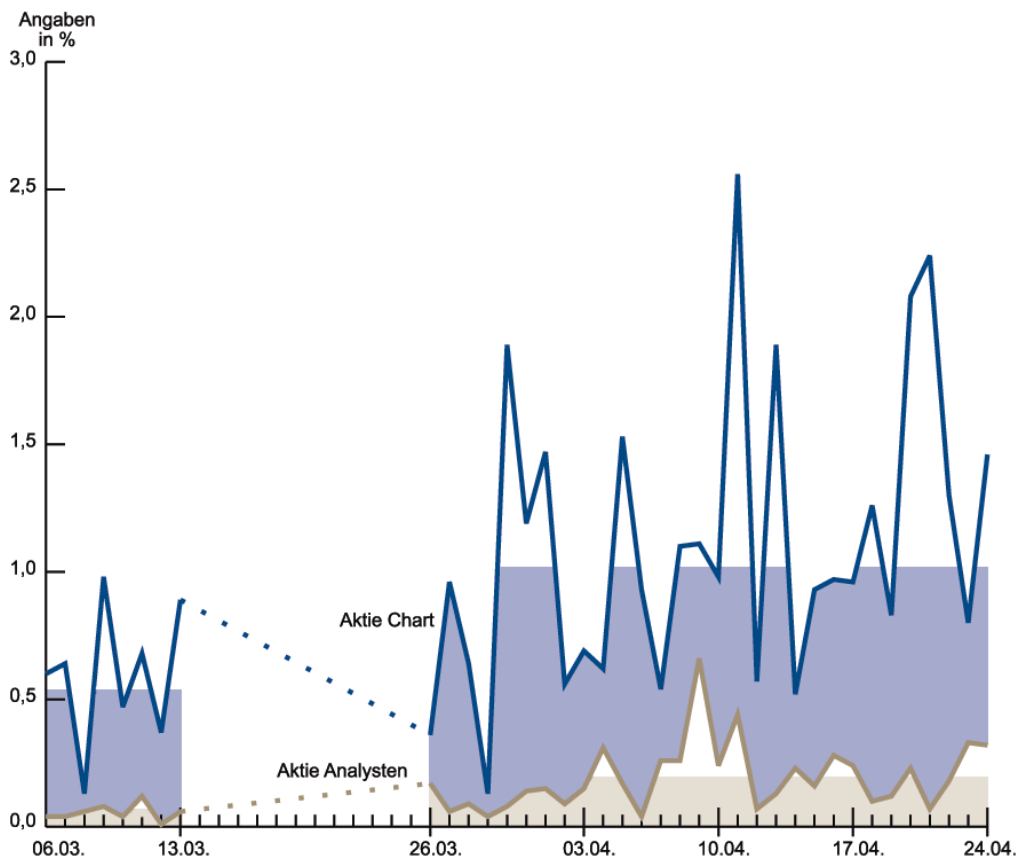


Abb. 4–16: Seitenaufrufe über Lesezeichen im Aktien-Bereich

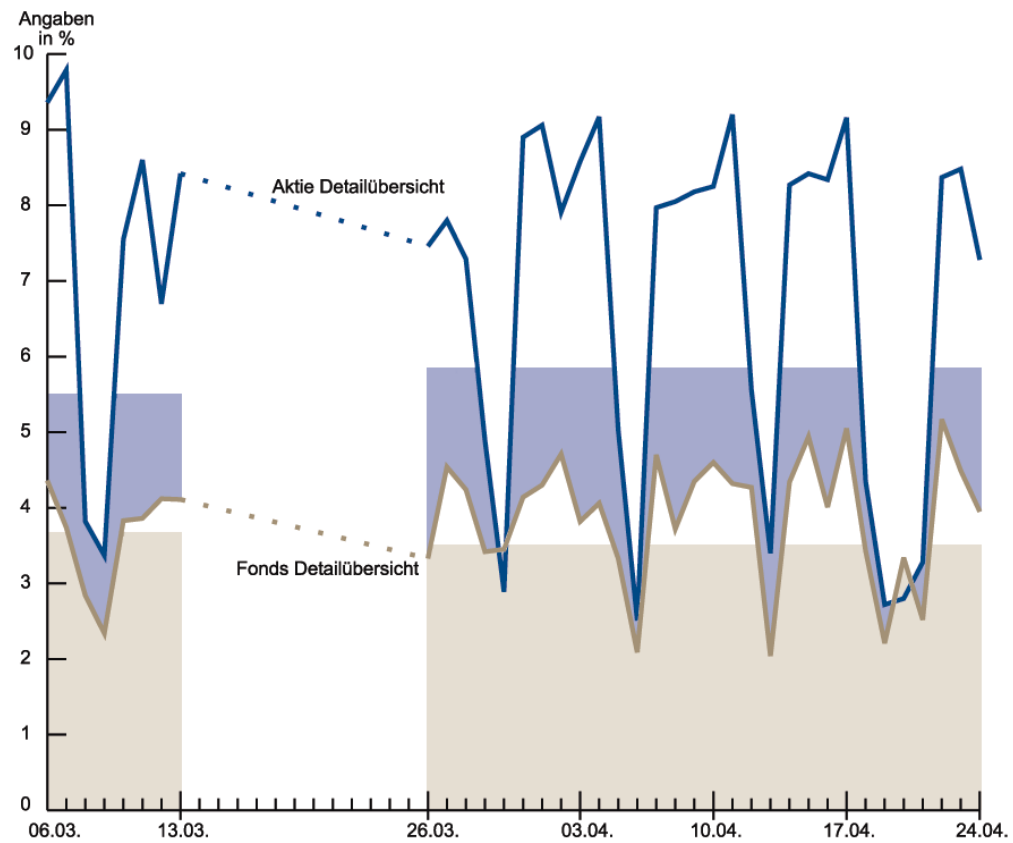


Abb. 4—17: Seitenaufrufe über Lesezeichen auf anderen Detailseiten

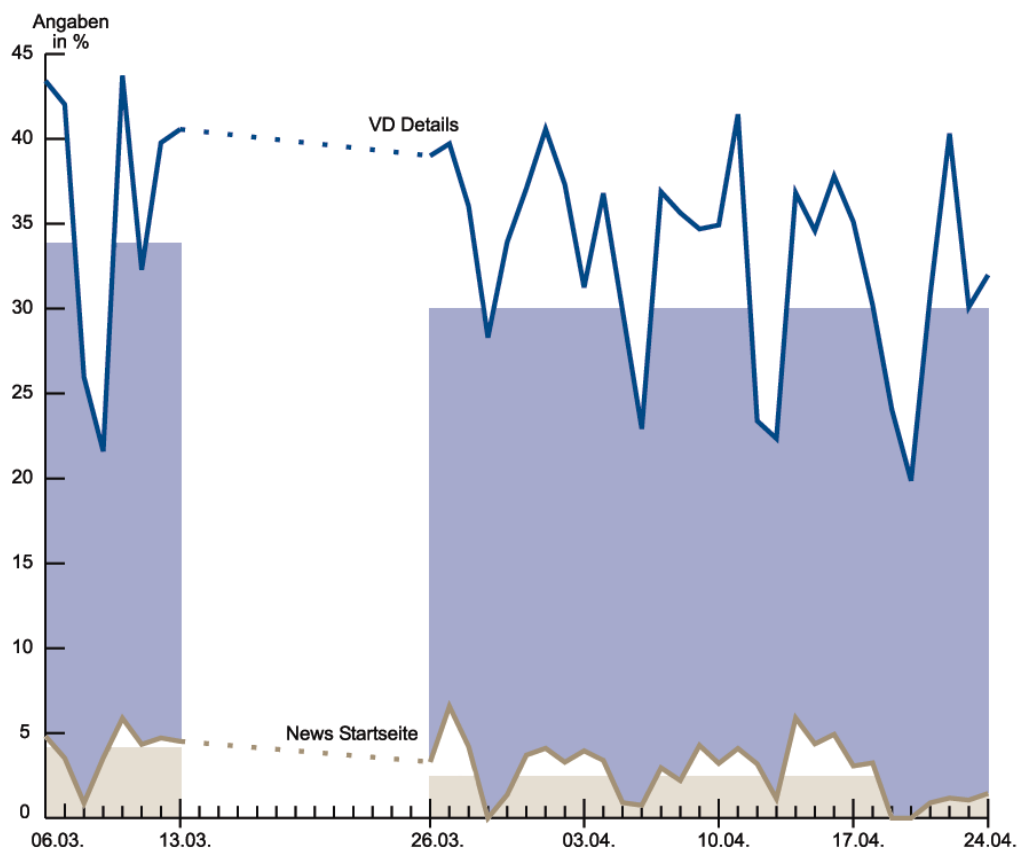


Abb. 4—18: Abnehmende Verwendung von Lesezeichen

In Tabelle 4–19 sind die Seiten aufgelistet, bei denen die Anteile an Aufrufen über Lesezeichen am höchsten sind. Zur besseren Übersicht sind an dieser Stelle nur die Mittelwerte für Vorher- und Nachher-Zeitraum mit dazugehörigem Wert von p angegeben. Eine vollständige Liste der tagesgenauen Beobachtungsdaten ist dem Anhang beigelegt (siehe Abschnitt 6.3 Auswertungsdaten, S. 71). In der Tabelle sind die 4 Seiten farblich hervorgehoben, für die der Anstieg der Aufrufe über Lesezeichen statistisch signifikant ist.

	Mittelwert		p
	vorher	nachher	
Ad hoc Meldungen Übersicht	3,99	1,51	0,6521
Aktie Analysten	0,06	0,19	0,0539
Aktie Chart	0,53	1,01	0,0021
Aktie Detailübersicht	5,48	5,82	0,7402
Aktie Kennzahlen	0,35	0,44	0,3922
Aktie News	0,01	0,30	0,1352
Aktien Startseite	1,30	1,19	0,1268
Analysten Übersicht	0,66	0,42	0,6030
AnleiheFinder - Ergebnis	0,05	0,38	0,7918
Anleihen Detailübersicht	0,63	1,00	0,0296
Fonds Detailübersicht	3,65	3,49	0,1928
Index Übersicht (Deu)	2,41	3,13	0,0418
Index Übersicht (Euro)	0,39	0,80	0,0090
Index Übersicht (Welt)	1,99	3,53	0,2914
News Startseite	4,03	2,38	0,0243
Optionsschein Detail	2,04	3,09	0,7274
Startseite Virtuelles Depot	0,64	1,47	0,1152
Übersicht Index-Zertifikat	2,38	1,84	0,0071
Übersicht sonstiges Zertifikat	1,57	1,96	0,4068
VD Details	33,74	29,86	0,0275
VD Übersicht Statistik	11,86	16,88	0,0005
Watchlist	13,12	21,86	0,0002

Tab. 4–19: Tatsächliche Verwendung von Lesezeichen (Übersicht)

4.4. Bewertung der Ergebnisse

Es wurden wegen des Lotsen pro Tag mehr Lesezeichen angelegt als vorher. Die Steigerung durch den Lotsen ist zwar rückläufig (am ersten Tag des Lotsen wurden über 20mal mehr Lesezeichen angelegt als vorher, Ende April »nur« noch 3- bis 4mal soviel), doch dies muss auch so sein. Nutzer, die vom Lotsen zur Verwendung von Lesezeichen angeregt wurden, legen typischerweise alle nötigen Lesezeichen auf einmal an. Danach werden die Lesezeichen genutzt, ohne dass neue hinzukommen.

Betrachtet man die aktiven Nutzer von *maxblue*, so verwenden durch den Lotsen erheblich mehr Nutzer als vorher Lesezeichen. Die entsprechende Prozentzahl hat sich nahezu verdoppelt. Dabei zeigten die verzögerten Explorationskarten in Popup-Fenstern die größte Wirkung, während die E-Mails des Lotsen kaum zum Anstieg beitragen. Ein direkter Vergleich von verzögerten Explorationskarten mit den E-Mails ist jedoch schwierig, da die genauen Nutzerzahlen der Explorationskarten aus technischen Gründen nicht ermittelt werden konnten.

Von den E-Mail-Empfängern hatte ein größerer Teil bereits Lesezeichen angelegt, als es für die aktiven Nutzer von *maxblue* charakteristisch war. Die E-Mails wurden also von einer besonderen Nutzergruppe verwendet, ohne dass hier auf die genauen Hintergründe eingegangen werden kann.

Einige Seiten von *maxblue* werden nun statistisch signifikant häufiger mit Lesezeichen aufgerufen als vor dem Lotsen. Dazu zählen die besonders häufig aufgerufenen Seiten aus dem geschlossenen Bereich.

Die geringe Wirkung der E-Mails legt die Vermutung nahe, dass die Distanz bei dieser Ausgabeform zu groß war. Da E-Mails in dieser Studie für eine besondere Nutzergruppe interessant waren, scheint die Wirkung dieser Ausgabeform anwendungsspezifisch zu sein. Die Aussagen können daher nicht einfach verallgemeinert werden.

Die Wirkung der verzögerten Explorationskarten zeigt, dass verzögerte Nutzungshinweise prinzipiell geeignet sind, um Lernerfolge zu erzielen. Dabei muss jedoch beachtet werden, dass die Verzögerung nicht zu groß wird.

5. Zusammenfassung und Ausblick

»Aus ergonomischer Sicht ist die Welt ein Jammertal.« [Heise 2002]
Hans Günter Siebert vom TÜViT über den Nachholbedarf von
Softwarefirmen bezüglich Gebrauchstauglichkeit

Die grundlegenden Überlegungen zum Erlernen von Softwareanwendungen (Kapitel 2, S. 5ff) resultieren in Anforderungen an das Anwendungsdesign, um kontinuierliches Weiterlernen zu erleichtern. Eine interaktive, nutzergeführte Softwareanwendung sollte sich dazu an den Prinzipien des Minimalismus orientieren. Es sollte Gelegenheiten zum aktiven Lernen geben. Hinweise und Erklärungen müssen sich an den Aufgaben der Nutzer orientieren, kurz gefasst sein und unterschiedliches Vorwissen berücksichtigen. Die Anwendung sollte explorationsfreundlich gestaltet sein und die intrinsische Motivation der Nutzer fördern. Die Benutzungsschnittstelle sollte möglichst wenig Modi enthalten und darf nur einfaches adaptives Verhalten aufweisen, bei dem kein Nutzermodell angelegt wird.

Die Analyse vorhandener Lösungen zur Anwenderunterstützung (Kapitel 3, S. 19ff) hat ergeben, dass sich bisherige Systeme nur wenig oder gar nicht zur Erleichterung kontinuierlichen Weiterlernens eignen. Obwohl die Prinzipien des Minimalismus schon sehr lange bekannt sind – wegweisend war bereits [Carroll 90] – werden sie in den derzeit existierenden Anwendungen kaum berücksichtigt. Eine Ausnahme bilden nur jene Ansätze, die explizit für eine Explorationsunterstützung gestaltet sind.

Anhand der Implementierung des *maxblue* Lotsen (Kapitel 4, S. 43ff) wurde untersucht, welche Wirkung Erklärungen haben, die zur Unterstützung der Nutzer zeitverzögert ausgegeben werden. Dabei wurde festgestellt, dass solche Ansätze effektiv sein können, wenn man sich um das richtige Maß an Verzögerung bemüht.

Zu der Frage, ob interaktive Anwendungen adaptierbar sein sollten, kann man sich die Lesezeichen von *maxblue* ansehen, die so eine Adaptionsmöglichkeit bieten. Der hohe Anstieg der Anzahl von Nutzern, die wegen des Lotsen nun Lesezeichen angelegt haben, deutet darauf hin, dass Nutzer Adaptionsmöglichkeiten gern verwenden, wenn sie deren Vorteile erkennen.

Computer wurden lange Zeit vor allem als Maschine, als Werkzeug oder auch als Kommunikationsmedium angesehen. Erst in den letzten Jahren setzt sich langsam das Leitbild des Computers als

Assistenz durch (vgl. [Donker 2002], [Krämer et al. 2002]). Da Wissen prinzipiell als nicht vermittelbar gilt, kann den Nutzern nur eine Unterstützung und Anregung gegeben werden, damit sie ihr Wissen über eine Anwendung ausbauen. Zu dieser Unterstützung, die für die Erleichterung kontinuierlichen Weiterlernens kennzeichnend ist, passt das Leitbild des Computers als Assistenz am besten.

Sehr interessant ist in diesem Zusammenhang trotz aller Kritik ein intelligentes Hilfesystem wie der Microsoft Office-Assistent. Sein Vorgänger, der Lumière-Prototyp, war ein viel versprechender Ansatz. Für alle negativen Eigenschaften des Office-Assistenten kann man im Lumière-Prototypen Lösungsvorschläge finden. Eine Unterstützungsform dieser Art ist zwar sehr aufwändig zu implementieren und daher nur für entsprechend große Softwareprojekte interessant, aber ein System, das den Nutzer, seine Fähigkeiten und aktuellen Bedürfnisse so genau analysiert, könnte für den Nutzer zu einem wertvollen Assistenten werden.

Für andere Projekte, die auf einfachere Lösungsvorschläge angewiesen sind, ist es interessant, die Möglichkeiten von verzögerten Erklärungen näher zu untersuchen. Dies konnte im Rahmen dieser Diplomarbeit nur in groben Zügen geschehen. Da sie bei *maxblue* anscheinend von einer untypischen Nutzergruppe verwendet wurden, wäre beispielsweise zu analysieren, unter welchen Umständen und für welche Nutzergruppen E-Mails das geeignete Medium für Nutzungshinweise sind. Als sehr effektiv haben sich in dieser Studie verzögerte Explorationskarten erwiesen. Die einfache Umsetzung macht diese Ausgabeform für viele Anwendungsgebiete interessant.

Ganz allgemein ist aber die in der Einleitung beschriebene zu hohe Komplexität von Computeranwendungen, Handys oder auch Videorecordern abhängig davon, ob der Gebrauchstauglichkeit der angemessene Stellenwert eingeräumt wird.

6. Anhang

6.1. Screenshots des *maxblue* Lotsen

Die Homepage weist den Nutzer in der Box »Ihre Vorteile« auf den *maxblue* Lotsen hin.

The screenshot shows the maxblue website homepage in Microsoft Internet Explorer. The browser address bar displays <http://www.maxblue.de/>. The page features a navigation bar with links: Home | Märkte | Analyse | Training | Produkte | Service & Dialog | Community. The main content area is divided into several sections:

- MY MAXBLUE**: A registration and login section with fields for Username and Passwort, and buttons for 'Registrieren' and 'Login'.
- IHRE VORTEILE**: A section titled 'Was ist maxblue?' with a list of tips and a 'Jetzt informieren!' button.
- KUNDE WERDEN**: A section featuring a woman's portrait and text about top investment know-how.
- FINANCIAL TIMES NEWS**: A section with a date '31.03.2003 / 13:47' and a list of news items, including 'SARS: Hongkong stellt Hunderte unter Zwangsquarantäne'.
- MARKTINFORMATIONEN**: A section with a search bar for 'Name, Ticker, WKN', a table of market indices, and a line chart showing the DAX index.
- DEUTSCHE BANK**: A section titled 'Deutsche Bank Privat- und Geschäftskunden AG' with a 'db PIP-SERVICE' sub-section.

The footer contains links for 'Site Map | AGBs | Presse | Disclaimer | Impressum | Kontakt' and copyright information: '© 2001 - 2003 | Deutsche Bank AG | Börsenplatz 7-11 | D-60272 Frankfurt am Main'.

Abb. 6—1: Homepage von *maxblue* (komplett)

Ebenso verweist die Anmelde-Seite unter »Tipps & Tricks« auf den *maxblue* Lotsen.

The screenshot shows the login page of the *maxblue* website. The browser window is titled "maxblue - Online handeln wie die Profis - Microsoft Internet Explorer". The address bar contains "http://www.maxblue.de/". The page layout includes a blue header with the *maxblue* logo and the Deutsche Bank logo. Below the header, there are navigation links: "Home | Märkte | Analyse | Training | Produkte | Service & Dialog | Community". A secondary navigation bar features buttons for "Kursabfrage", "Kunde werden", and "zum Depot". The main content area is titled "My maxblue - Ihr kostenloses Serviceangebot" and contains a paragraph: "Informieren Sie sich schnell und professionell mit Realtimekursen, Watchlist und dem Research der Deutsche Bank Experten. Nutzen Sie darüber hinaus professionelle Tools wie RisikoKompass und vieles mehr." Below this, there are three main sections: "Registrierung" (Registration), "Log-In", and "Ihre Vorteile" (Your Advantages). The "Registrierung" section includes a sub-section "Sind Sie neu hier?" with a "Registrieren" button. The "Log-In" section has a form with "Username" and "Password" fields and a "Log-In" button. The "Ihre Vorteile" section lists services like "E-Mail-Registrierung", "Adressregistrierung", and "Kundenregistrierung". A "Depot" section is also visible. At the bottom, there is a footer with "Site Map | AGBs | Presse | Disclaimer | Impressum | Kontakt" and copyright information for Deutsche Bank AG.

Abb. 6—2: Anmelde-Seite von *maxblue*

Auch die Startseite von »My maxblue«, wo die Nutzer ihre persönlichen Daten und Einstellungen verwalten, verweist auf den *maxblue* Lotsen. Diese Seite können Nutzer nur nach einer Anmeldung sehen.

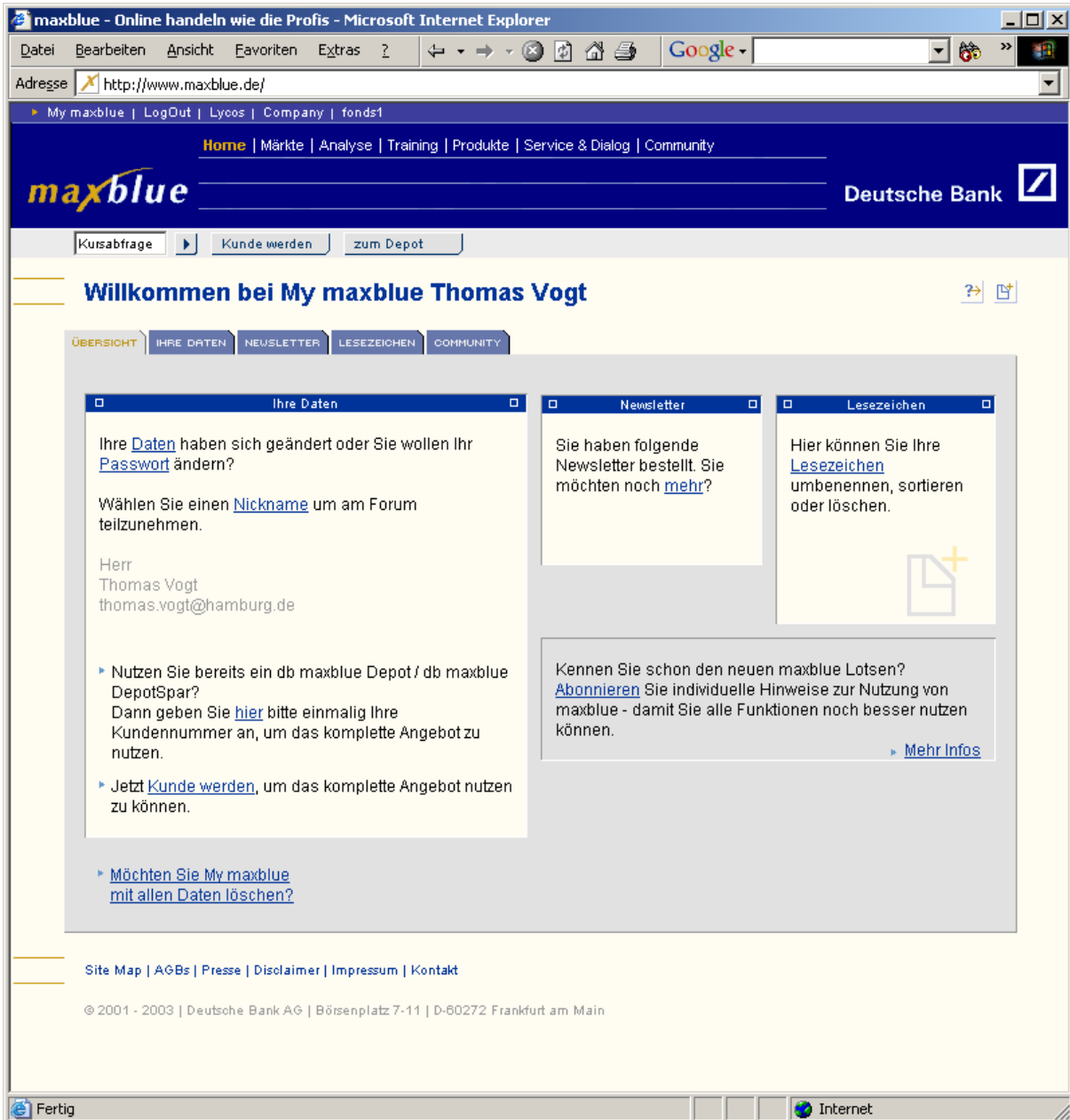


Abb. 6—3: »My maxblue« Startseite

Wenn der Nutzer »mehr Infos« wählt, wird ihm in einem Popup-Fenster eine reine Informationsseite angezeigt.

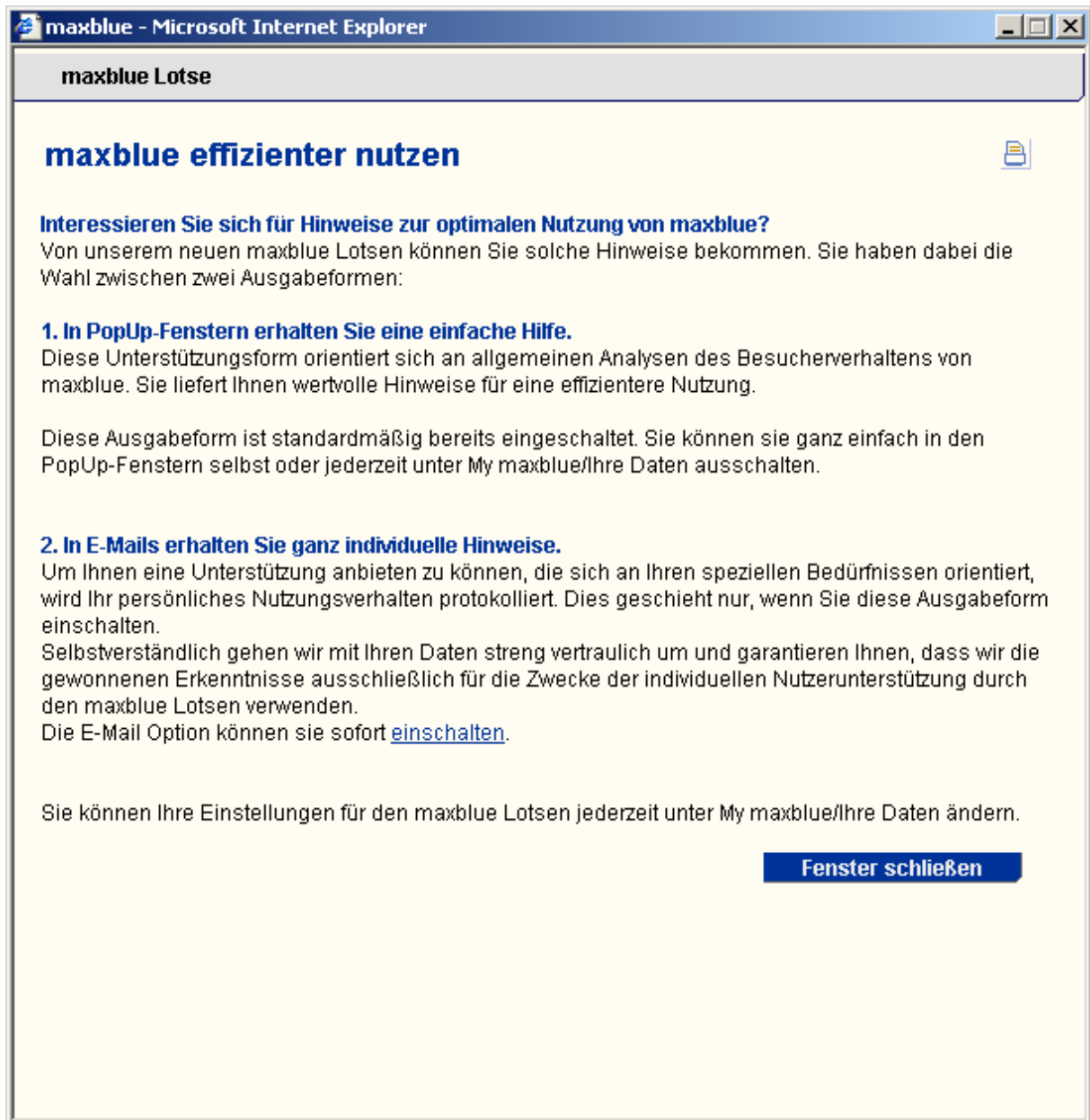


Abb. 6—4: Informationsseite des Lotsen

Wählt der Nutzer »Einschalten« auf der Informationsseite oder bereits auf einer der vorherigen Seiten, ohne angemeldet zu sein, muss er dies nachholen.

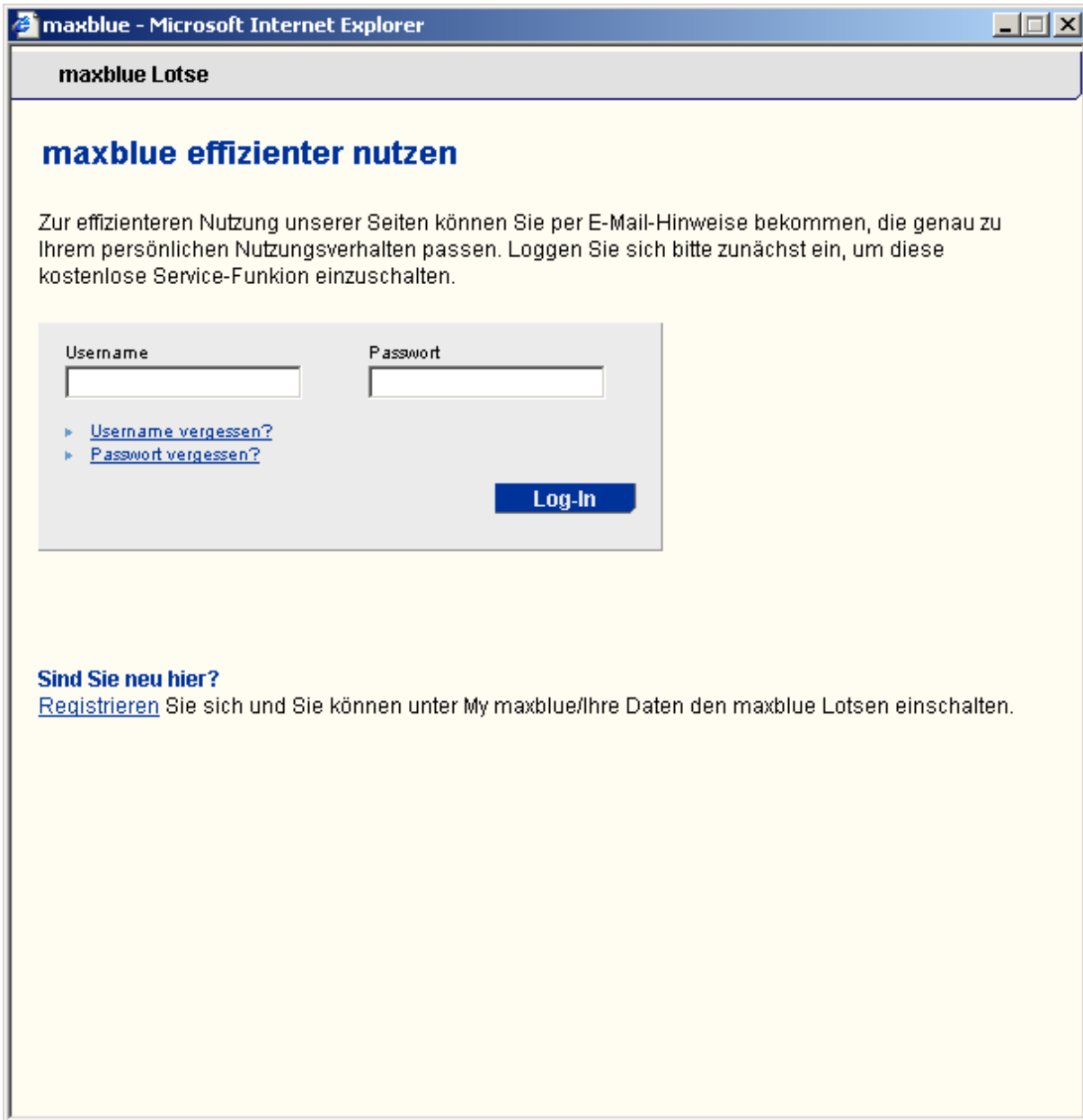


Abb. 6—5: Spezielle Anmelde-Seite für den Lotsen

Sonst kommt der Nutzer ohne den letzten Zwischenschritt direkt zum Einschalten der E-Mail-Option.

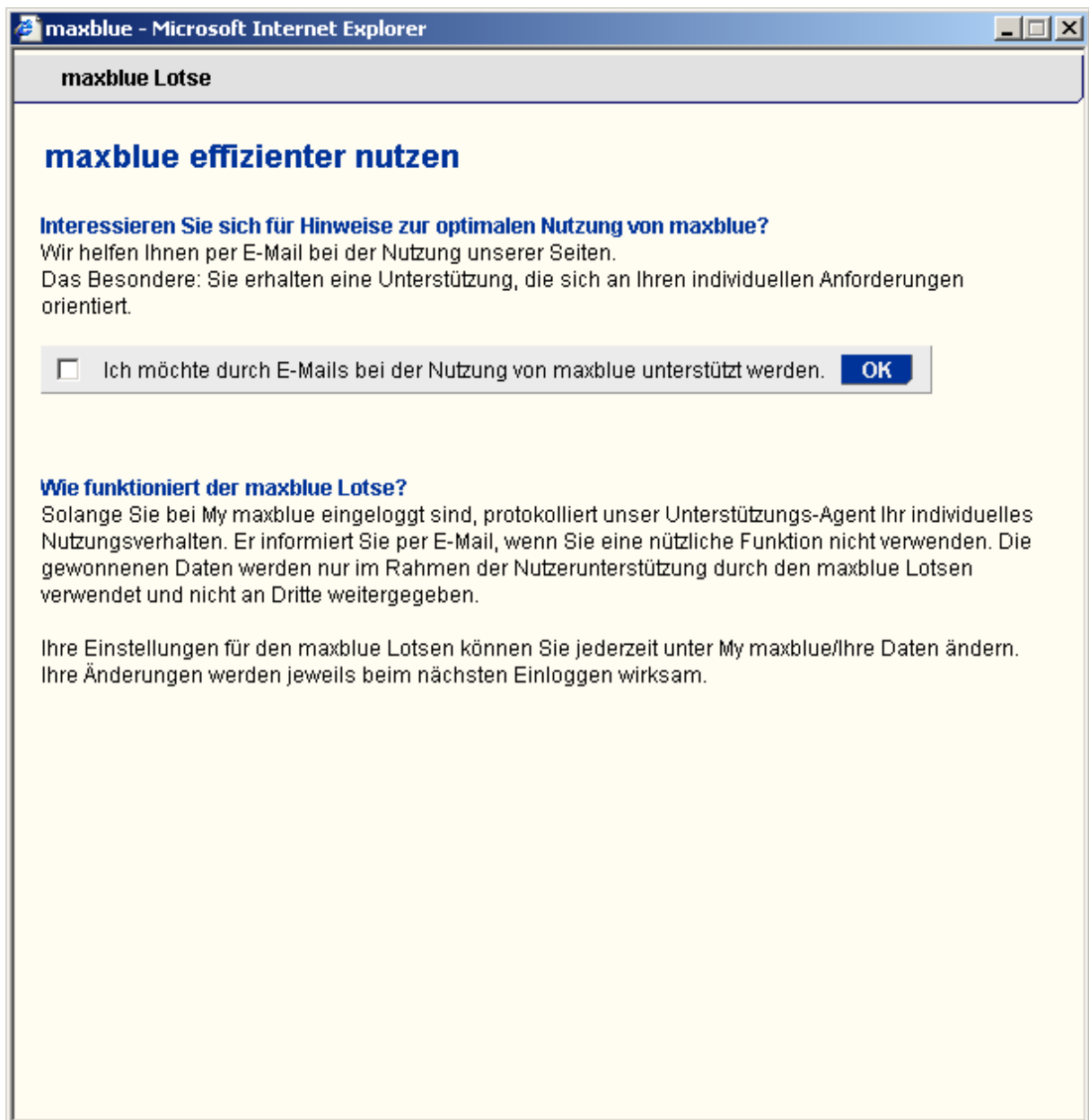


Abb. 6—6: Einschalten der E-Mail-Option des Lotsen

Auf der Seite »Ihre Daten« kontrolliert der Nutzer die E-Mail- und die Popup-Option. Ebenso kann er sich hier jederzeit die Informationsseite ansehen.

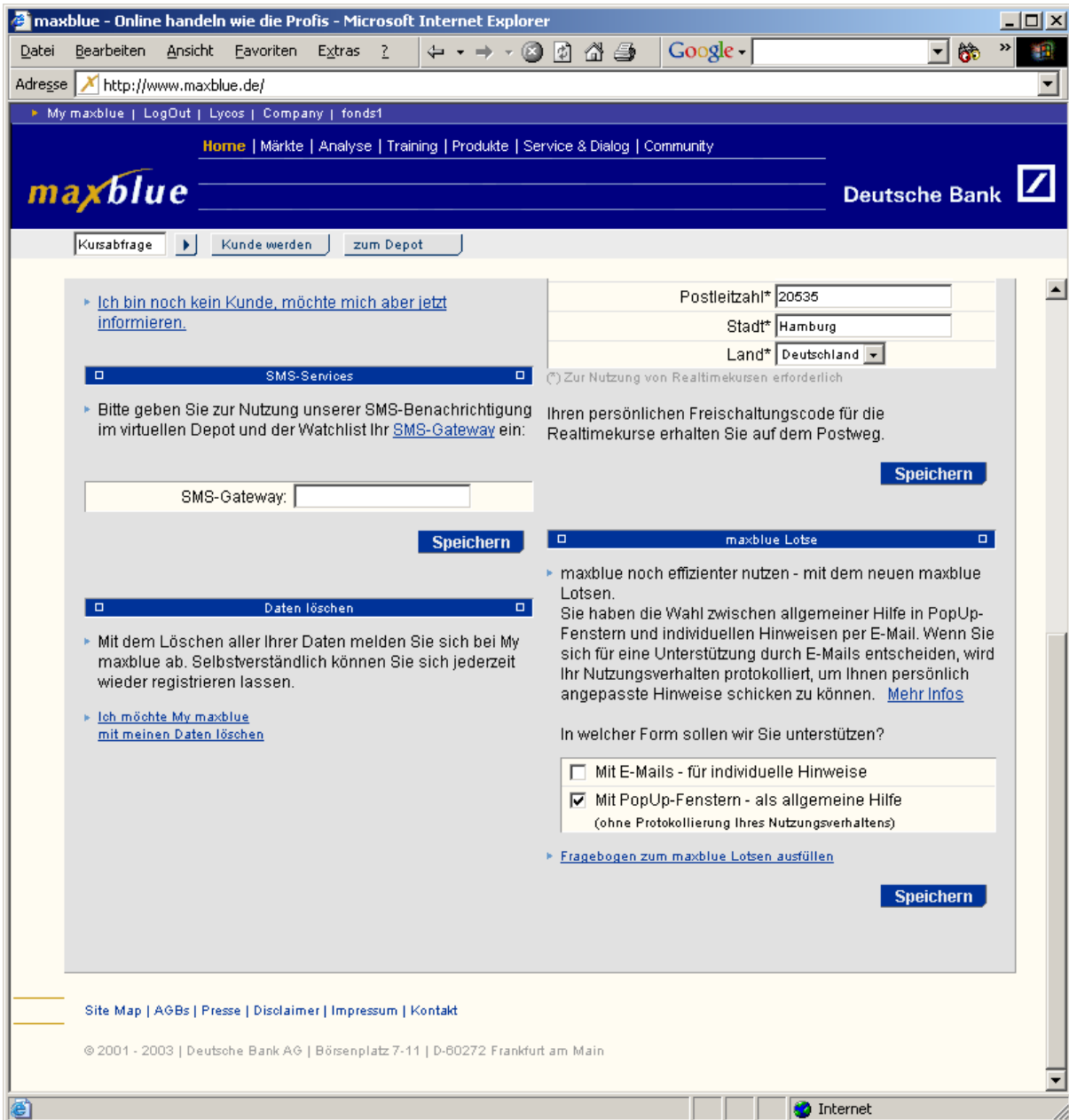


Abb. 6—7: Verwaltung der Einstellungen zum Lotsen

6.2. Beispiel einer vom Lotsen versendeten E-Mail

6.2.1. Text der E-Mail

An: karl@mustermann.de
Von: lotse@maxblue.de
Betreff: Lesezeichen für direkten Zugriff auf wichtige Services
Anlagen: lotse.pdf (64 KB)

Hallo Herr Mustermann,

heute hat der maxblue Lotse einen Hinweis zu den Lesezeichen für Sie parat:

Mit den Lesezeichen greifen Sie direkt auf die maxblue Services zu, die Ihnen am wichtigsten sind. Sobald Sie eingeloggt sind, profitieren Sie in wenigen Schritten von dieser Funktionalität:

1. Auf fast allen Seiten befindet sich ein Lesezeichen-Icon am rechten, oberen Seitenrand - zwischen dem Hilfe- und dem Druck-Icon.
2. Klicken Sie es an, wenn Sie sich eine Seite merken wollen, und geben Sie dem Lesezeichen einen eigenen Namen.
3. Nach Klick auf "Speichern" erscheint das Lesezeichen in der Leiste über dem maxblue Logo. Sie können es dort jederzeit anklicken, um direkt zur gewünschten Seite zu gelangen.

So ein Lesezeichen können Sie z.B. auch für Ihr "Virtuelles Depot" setzen.

Achten Sie beim Anlegen eines neuen Lesezeichens darauf, dass die Leiste mit den Lesezeichen noch nicht voll ist. Wenn Sie kürzere Namen vergeben, passen mehr Lesezeichen in die Leiste. Unter My maxblue/Lesezeichen können Sie jederzeit Ihre Lesezeichen umbenennen, neu sortieren oder löschen.

Das angehängte PDF-Dokument enthält eine illustrierte Erklärung zur Verwendung der Lesezeichen. Falls Sie keinen Acrobat Reader zum Anzeigen von PDF-Dokumenten haben, steht dieser Hinweis auch online für Sie bereit:

http://www.maxblue.de/?directJump=/ssag/reloader_mari0857.html


Diese E-Mail wurde Ihnen zugesandt, da Sie die E-Mail-Option des maxblue Lotsen eingeschaltet haben. Sie können diese Option jederzeit unter My maxblue/Ihre Daten wieder ausschalten.

Sie haben noch Fragen oder Anregungen? Bitte antworten Sie nicht direkt über diese E-Mail - wenn Sie die Reply-to-Funktion wählen, können wir Ihre Anfrage leider nicht bearbeiten. Wählen Sie einfach unter


<http://www.maxblue.de/index.html?jumpto=kontakt>

in welcher Form Sie den Kontakt zu uns aufnehmen möchten

6.2.2. Anhang der E-Mail




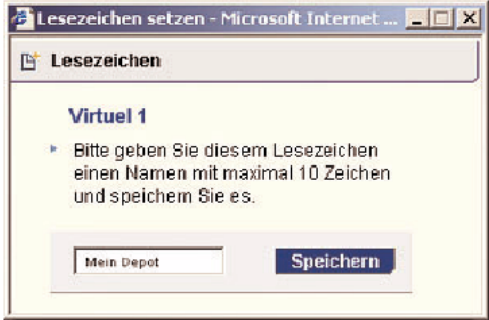

**maxblue Lotse
Lesezeichen**



Deutsche Bank
Privat- und Geschäftskunden AG

Kundenservice
53255 Bonn
Telefon: 0 18 18- 1000
Telefax: 0 18 18- 1001
Internet:
www.deutsche-bank.de
www.maxblue.de

Wussten Sie schon, dass ...?
... Sie mit der Lesezeichen-Funktion innerhalb von My maxblue direkten Zugriff auf von Ihnen gewählte Services haben? Es geht ganz einfach:

1. Sie rufen die Lesezeichen-Funktion mit Klick auf das Icon am rechten oberen Seitenrand auf.
 
2. Anschließend können Sie dem Lesezeichen einen von Ihnen gewählten Namen zuweisen.
 
3. Nach Klick auf "Speichern" erscheint das Lesezeichen in der Leiste über dem maxblue Logo.
 

Unter My maxblue/Lesezeichen können Sie jederzeit Ihre Lesezeichen umbenennen, neu sortieren oder löschen.

Probieren Sie es aus und überzeugen Sie sich selbst, wie nützlich die Lesezeichen für Sie sind!

Abb. 6—8: Angehängte PDF-Datei einer E-Mail des Lotsen

6.3. Auswertungsdaten

An dieser Stelle sind die tagesgenauen Beobachtungsdaten über die tatsächliche Verwendung von Lesezeichen aufgelistet. Die Interpretation der Daten befindet sich im Abschnitt 4.3 Evaluation des Lotsen (S. 48ff).

Für die einzelnen Seiten ist prozentual angegeben, wie häufig sie am jeweiligen Tag per Lesezeichen aufgerufen wurden. Farblich hervorgehoben sind die Seiten, zu denen es in den Abbildungen 4–14 bis 4–18 (S. 53ff) eine graphische Darstellung der Werte gibt.

	Ad hoc Meldungen	Übersicht	Aktie Analysten	Aktie Chart	Aktie Detailübersicht	Aktie Kennzahlen	Aktie News	Aktien Startseite	Analysten Übersicht	AnleiheFinder - Ergebnis	Anleihen Detailübersicht	Fonds Detailübersicht	Index Übersicht (Deu)	Index Übersicht (Euro)	Index Übersicht (Weit)	News Startseite	Optionsschein Detail	Startseite Virtuelles Depot	Übersicht Index-Zertifikat	Übersicht sonstiges Zertifikat	VD Details	VD Übersicht Statistik	Watchlist
März	06.	1,39	0,04	0,60	9,36	0	0	1,31	0,68	0	1,21	4,36	4,82	0,78	1,63	4,82	3,37	0,75	7,52	1,74	43,41	14,25	21,09
	07.	1,20	0,04	0,64	9,79	0	0	1,08	0,67	0,03	0,78	3,74	4,71	0,33	3,74	3,53	3,22	1,38	3,65	1,80	42,02	14,32	16,80
	08.	1,85	0,06	0,13	3,82	0,71	0	0,78	0,17	0,36	0,40	2,84	1,09	0,46	1,84	0,88	1,11	1,00	2,05	3,17	25,97	8,98	10,00
	09.	16,04	0,08	0,98	3,37	0	0	4,18	0	0	0,29	2,34	1,05	0	0,39	3,54	1,19	1,05	1,52	0,29	21,60	9,10	12,57
	10.	1,52	0,04	0,47	7,55	0,26	0	1,47	1,24	0	1,12	3,83	3,16	0,35	2,93	5,88	2,97	0,66	0,88	2,77	43,73	15,16	17,14
	11.	2,43	0,12	0,68	8,60	0,11	0	0,90	1,32	0	0,92	3,86	3,20	1,00	2,50	4,34	2,70	0,55	0,74	1,86	32,29	12,27	19,45
	12.	2,36	0,01	0,37	6,70	0,10	0,03	1,35	1,30	0	0,52	4,12	3,77	0,73	2,17	4,72	1,45	0,64	0,92	2,08	39,77	14,78	16,81
	13.	6,17	0,06	0,89	8,42	1,68	0,03	1,20	1,04	0	1,20	4,11	3,90	0,49	2,98	4,52	2,71	0,50	1,76	0,49	40,56	14,17	16,12
	26.	1,36	0,17	0,36	7,46	0	0,14	1,29	1,19	0,14	1,58	3,33	2,28	0,69	6,88	3,31	3,24	1,36	1,14	1,67	39,00	15,07	19,12
	27.	1,64	0,06	0,96	7,80	0	0	2,71	0,90	0,06	1,40	4,54	2,50	1,67	9,57	6,58	2,53	1,34	1,19	3,38	39,71	19,55	25,13
	28.	1,01	0,09	0,64	7,29	0	0	2,15	0,91	0	0,94	4,24	2,24	0,66	4,73	4,20	7,17	1,56	3,23	4,25	36,02	17,31	22,82
	29.	2,63	0,04	0,13	4,89	0	0	0,74	0,34	0	0,77	3,42	2,29	2,16	1,02	0	2,80	0,53	1,68	2,33	28,30	11,33	15,75
	30.	0	0,08	1,89	2,89	0,32	0,14	0,91	0	0,06	0,36	3,45	4,27	1,18	0,60	1,37	0,60	0,98	4,08	0	33,91	11,54	15,93
31.	0,76	0,14	1,19	8,90	0	0,31	1,75	0,66	0	1,08	4,14	4,67	0,58	6,01	3,71	3,00	1,68	3,55	4,44	37,08	17,15	20,57	
April	01.	3,97	0,15	1,47	9,06	1,86	0,07	1,20	0,71	0	0,73	4,30	5,53	0,90	4,51	4,11	3,44	1,02	1,62	5,65	40,57	18,66	22,75
	02.	2,57	0,09	0,56	7,91	0	0,12	1,38	0,18	0	1,05	4,71	6,11	0,88	4,54	3,29	4,79	1,45	1,52	3,02	37,31	20,04	23,41
	03.	1,47	0,15	0,69	8,58	1,12	0,39	1,56	0,50	1,26	0,48	3,82	4,95	1,24	8,04	3,96	3,14	1,23	0,68	1,34	31,25	20,26	25,27
	04.	2,30	0,31	0,62	9,17	1,32	0,49	1,80	0,44	0	0,82	4,06	4,46	0,92	3,53	3,40	5,43	1,03	1,90	3,52	36,79	17,79	21,84
	05.	0	0,17	1,53	5,03	0,28	0,11	1,28	0,21	0,50	0,61	3,33	1,43	1,60	2,31	0,91	2,39	2,33	5,26	1,44	29,88	15,74	22,32
	06.	0	0,04	0,93	2,51	0,42	0,17	1,36	0,17	0,28	0,14	2,09	0,50	0	0,24	0,75	1,15	1,68	0	1,41	22,92	9,68	27,67
	07.	0	0,26	0,54	7,97	0,38	0,27	2,08	0,55	0,76	0,76	4,70	3,93	0,92	6,08	2,96	2,98	1,37	2,07	1,13	36,88	19,83	21,88
	08.	4,78	0,26	1,10	8,05	0,47	0,25	1,14	0,54	0,78	0,71	3,72	3,67	0,53	3,66	2,21	6,81	2,25	3,19	2,19	35,64	19,44	27,88
	09.	0,96	0,66	1,11	8,18	0,26	0,48	1,63	0,92	0,41	1,18	4,35	4,45	0,84	5,08	4,28	4,55	1,14	1,92	0,99	34,69	20,54	23,66
	10.	1,82	0,24	0,98	8,25	0,82	0,79	1,19	0,61	0,32	1,41	4,60	2,47	0,84	4,01	3,22	6,94	1,64	1,49	0,80	34,92	19,03	21,65
	11.	5,31	0,44	2,56	9,20	0,59	0,68	1,15	0,67	1,24	1,54	4,32	4,01	1,07	3,70	4,10	5,52	1,27	3,11	1,07	41,44	19,17	23,33
	12.	0	0,07	0,57	5,56	0,65	0,24	0,73	0	0,45	0,21	4,27	0,82	0	0,55	3,19	2,29	0,93	0	0	23,39	12,70	28,59
	13.	1,22	0,13	1,89	3,40	0,70	0,25	1,46	0,52	0	0,35	2,04	1,50	1,21	1,29	1,14	4,59	1,66	0,86	1,39	22,35	13,55	18,66
14.	0	0,23	0,52	8,27	0,98	0,66	1,65	0,46	0,74	2,31	4,34	2,56	1,03	2,31	5,89	5,30	1,33	1,81	0,95	36,82	21,97	20,51	
15.	1,94	0,16	0,93	8,42	0	0,60	1,90	0,62	0,85	1,61	4,94	3,94	0,78	5,47	4,37	4,65	2,02	3,13	1,30	34,59	21,93	23,32	
16.	2,01	0,28	0,97	8,34	0,41	0,62	1,46	0,52	0,53	2,03	4,01	3,53	1,18	4,56	4,93	2,53	2,01	3,63	1,37	37,80	22,26	25,80	
17.	1,89	0,24	0,96	9,16	0,44	0,82	1,22	0,76	1,17	1,53	5,05	4,22	0,41	3,49	3,07	4,32	1,85	1,89	1,10	35,11	19,69	23,54	
18.	2,94	0,10	1,26	4,35	1,06	0,16	0,96	0	0,15	0,54	3,43	2,22	0,96	2,87	3,25	4,56	2,46	1,01	0,97	30,22	17,56	22,72	
19.	0	0,12	0,83	2,72	0	0,13	1,21	0	0,84	0,97	2,21	3,32	1,50	0,24	0	1,94	1,05	1,65	0	24,06	14,53	15,12	
20.	0	0,23	2,08	2,80	0,38	0,32	0,39	0	0,15	1,00	3,34	2,52	0	0,94	0	1,07	2,23	0	1,85	19,87	12,07	22,49	
21.	0	0,07	2,24	3,28	0	0,12	0,22	0	0,71	0,52	2,52	0,98	0,37	1,33	0,90	1,59	1,23	0	0,85	30,86	12,41	29,38	
22.	2,05	0,18	1,30	8,37	0,60	0,14	1,02	0,32	0,40	2,42	5,17	3,86	0,93	4,54	1,18	3,67	1,66	2,26	2,73	40,31	18,89	23,37	
23.	0,85	0,33	0,80	8,48	1,00	0,67	0,98	0,51	0,18	1,35	4,49	4,91	0,97	3,70	1,06	4,00	1,30	1,64	9,38	30,09	19,01	22,28	
24.	3,76	0,32	1,46	7,28	1,29	0,32	1,53	0,77	0,29	1,78	3,95	3,88	1,49	7,17	1,45	4,10	1,68	1,52	1,22	31,98	19,06	22,35	

Tab. 6—9: Tatsächliche Verwendung von Lesezeichen

6.4. Literaturverzeichnis

[Adams et al. 2001]

Wendy Adams, Jiraphan Brown, Dorothy Rapeepun, Wayne Williams. The Effectiveness of Online Help Systems: Text Only, Animated Images Only, and Integrated Interactive. University of Maryland, 2001. http://www.otal.umd.edu/SHORE2001/help/online_help.PDF (zuletzt besucht am 11.04.2002)

[Apple 2002]

Inside Mac OS X: Aqua Human Interface Guidelines. Apple Computer, Inc., 2002. <http://developer.apple.com/techpubs/macosx/Essentials/AquaHIGuidelines/AquaHIGuidelines.pdf> (30.09.2002)

[Arar o.J.]

Yardena Arar. Bob Strives to Make New Strides in User-Friendliness. In: Los Angeles Daily News. o.J. <http://is.gseis.ucla.edu/impact/w95/RN/apr14news/Merc-news-bob.html> (07.11.2002)

[Bartz 2001]

Stephan Bartz. Gestaltungsvarianten des Usability-Testing. Wirtschafts- und Sozialwissenschaftliche Fakultät der Universität zu Köln, 2001. <http://www.systementwicklung.uni-koeln.de/forschung/veroeffentlichungen/dokumente/diplomarbeit-bartz.pdf> (11.04.2002)

[Bente/Krämer 2000]

Gary Bente, Nicole C. Krämer. Virtuelle Gesprächspartner: Psychologische Beiträge zur Entwicklung und Evaluation anthropomorpher Schnittstellen. In: Multimodale Interaktion im Bereich der Prozessführung. 42. Fachausschußsitzung Anthropotechnik, DGLR-Bericht 2000-02. Hrsg.: K. P. Gärtner. Deutsche Gesellschaft für Luft- und Raumfahrt, Bonn, 2000, S. 29-50. <http://www.embassi.de/publi/veroeffent/evaINP004KHM01.pdf> (30.08.2002)

[Bente/Krämer 2001]

Gary Bente, Nicole C. Krämer. Psychologische Aspekte bei der Implementierung und Evaluation nonverbal agierender Interface-Agenten. In: Mensch & Computer 2001: 1. Fachübergreifende Konferenz. Hrsg.: Horst Oberquelle, Reinhard Oppermann, Jürgen Krause. B. G. Teubner, Stuttgart, 2001, S. 275-285. <http://www.embassi.de/publi/veroeffent/phhINP045KHM01.pdf> (30.08.2002)

[BildscharbV 96]

Verordnung über Sicherheit und Gesundheitsschutz bei der Arbeit an Bildschirmgeräten (BildscharbV). Bundesministerium für Arbeit und Sozialordnung, 1996. http://www.bma.de/doc/doc_request.cfm?48E549BC94FF4D4394A9EB805777D10B (26.04.2002)

[Brodbeck/Rupietta 94]

Felix C. Brodbeck, Walter Rupietta. Fehlermanagement und Hilfesysteme. In: Einführung in die Software-Ergonomie – Gestaltung graphisch-interaktiver Systeme: Prinzipien, Werkzeuge, Lösungen. Hrsg.: Edmund Eberleh, Horst Oberquelle, Reinhard Oppermann. Walter de Gruyter, Berlin u.a., 1994, S. 197-234.

[Carroll 90]

John M. Carroll. The Nurnberg funnel: designing minimalist instruction for practical computer skill. MIT Press, Cambridge, Massachusetts u.a., 1990.

[Carroll 96]

John M. Carroll. Becoming social: expanding scenario-based approaches in HCI. In: Behaviour & Information Technology, 1996, Vol. 15, No. 4. 1996, S. 266-275. <http://people.cs.vt.edu/~carroll/papers/BecomingSocial-BIT96.pdf> (11.04.2002)

[Carroll/Aaronson 88]

John M. Carroll, Amy P. Aaronson. Learning by doing with simulated intelligent help. In: Communications of the ACM, Vol. 31, No. 9. 1988, S. 1064-1079.
<http://people.cs.vt.edu/~carroll/papers/SimulatedHelp-CACM88.pdf> (11.04.2002)

[Carroll/van der Meij 98]

John M. Carroll, Hans van der Meij. Ten Misconceptions about Minimalism. In: Minimalism Beyond the Nurnberg Funnel. Hrsg.: John M. Carroll. MIT Press, Cambridge, Massachusetts u.a., 1998, S. 55-90.

[Covi/Ackerman 95]

Lisa M. Covi, Mark S. Ackerman. Such Easy-to-Use Systems! How Organizations Shape the Design and Use of Online Help Systems. In: Proceedings of COOCS '95 Conference on Organizational Computing Systems, San Jose, CA. 1995. <http://scils.rutgers.edu/~covi/doc/coocs95.txt> (23.04.2002)

[DATech 2001]

DATech-Prüfhandbuch Gebrauchstauglichkeit. DATech Deutsche Akkreditierungsstelle Technik e.V., Frankfurt/Main, 2001. http://www.datech.de/download/deutsch/01-2002_Handbuch_Gebrauchstauglichkeit_V3.2.pdf (28.08.2002)

[Deimann 2002]

Markus Deimann. Motivationale Bedingungen beim Lernen mit Neuen Medien. In: Medienunterstütztes Lernen – Beiträge von der WissPro-Wintertagung 2002. Hrsg.: Wolf-Gideon Bleek, Detlev Krause, Horst Oberquelle, Bernd Pape. 2002, S. 61-70. http://www.wisspro.de/papers/fbi_bericht_bleek_etal_medienunterstuetztes_wintertagung_2002.pdf (09.08.2002)

[Dietz/Lang 99]

Richard B. Dietz, Annie Lang. Effective Agents: Effects of Agent Affect on Arousal, Attention, Liking & Learning. In: Third International Cognitive Technology Conference, CT'99, 11.-14. August 1999, San Francisco/Silicon Valley, USA. 1999. <http://www.cogtech.org/CT99/Dietz.htm> (06.09.2002)

[Donker 2002]

Hilko Donker. Didaktisches Interaktions-und Informationsdesign. dissertation.de – Verlag im Internet GmbH, Berlin, 2002. <http://www.dissertation.de/PDF/hd519.pdf> (06.09.2002)

[Eco 85]

Umberto Eco. Wie man Gebrauchsanweisungen befolgt. In: Wie man mit einem Lachs verweist – und andere nützliche Ratschläge. Deutscher Taschenbuch Verlag, München, 1999, S. 45-48.

[Encarnaçao 97]

L. Miguel Encarnaçao. Concept and Realization of Intelligent User Support in Interactive Graphics Applications. Dissertation, Eberhard-Karls-Universität zu Tübingen, Fakultät für Informatik, 1997. <http://www.crcg.edu/company/staff/mencarna/pubs/diss/disso.html> (26.09.2002)

[EU 90]

Richtlinie 90/270/EWG. Europäische Union, 1990. http://europa.eu.int/smartapi/cgi/sga_doc?smartapi!celexapi!prod!CELEXnumdoc&lg=DE&numdoc=31990L0270&model=guichett (26.04.2002)

[Farkas 98]

David K. Farkas. Layering as a Safety Net for Minimalist Documentation. In: Minimalism Beyond the Nurnberg Funnel. Hrsg.: John M. Carroll. MIT Press, Cambridge, Massachusetts u.a., 1998, S. 247-274.

[Fischer/Scharff 98]

Gerhard Fischer, E. Scharff. Learning Technologies in Support of Self-Directed Learning. In: Journal of Interactive Media in Education, 1998 (4). <http://www-jime.open.ac.uk/98/4/fischer-98-4.pdf> (11.04.2002)

[Frith 97]

Constance Frith. Motivation To Learn. 1997. <http://www.usask.ca/education/coursework/802papers/Frith/Motivation.PDF> (02.08.2002)

[Göbel 2001]

Matthias Göbel. Lehrunterlagen Angewandte Daten- und Informationsverarbeitung WS 2001/2002 – Hilfesysteme. Technische Universität Berlin – Lehrstuhl für Arbeitswissenschaft und Produktergonomie, 2001, Lehrunterlagen. http://www.awb.tu-berlin.de/Studium/Lehrunterlagen_und_Pruefungsleistungen/Daten-_und_Info-Verarbeitung/4._Benutzungsoberflaechengestaltung/4.3_Hilfesysteme/_uebersicht.html (20.03.2003)

[Graesser/Murray 90]

Arthur C. Graesser, Kelly Murray. A Question-Answering Methodology for Exploring a User's Acquisition and Knowledge of a Computer Environment. In: Cognition, Computing, and Cooperation. Hrsg.: Scott P. Robertson, Wayne Zachary, John B. Black. Ablex, Norwood, New Jersey, 1990, S. 237-267.

[Gronover/Senger/Riempp 2002]

Sandra Gronover, Enrico Senger, Gerold Riempp. Management multimedialer Kundeninteraktionen – Grundlagen und Entscheidungsunterstützung. In: i-com, Zeitschrift für interaktive und kooperative Medien, Heft 1-2002, S. 25-31.

[Hackos 98]

JoAnn T. Hackos. Choosing a Minimalist Approach for Expert Users. In: Minimalism Beyond the Nurnberg Funnel. Hrsg.: John M. Carroll. MIT Press, Cambridge, Massachusetts u.a., 1998, S. 149-177.

[Heckerman/Horvitz 98]

David Heckerman, Eric Horvitz. Inferring Informational Goals from Free-Text Queries: A Bayesian Approach. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, July 1998. Morgan Kaufmann, San Francisco, 1998, S. 230-237. <ftp://ftp.research.microsoft.com/pub/ejh/aw.pdf> (13.09.2002)

[Heinecke 98]

Andreas M. Heinecke. Interaktive multimediale Systeme – Teil 1: Mensch-Maschine-Interaktion. 1998, Lehrunterlagen. http://www.drheinecke.de/fh_do/files/ims1.pdf (20.03.2003)

[Heise 2002]

Felix Rehwald. Am Nutzer vorbei programmiert. Heise News-Ticker vom 9.12.2002. <http://www.heise.de/newsticker/data/jk-09.12.02-001/> (16.06.2003)

[Heise 2003]

n.a. Funktionsvielfalt und Bedienung bei modernen Handys schrecken Benutzer ab. Heise News-Ticker vom 8.6.2003. <http://www.heise.de/newsticker/data/jk-08.06.03-000/> (15.06.2003)

[Hochhaus o.J.]

Stephan Hochhaus. Motivation. o.J. <http://www.lernen-heute.de/motivation.html> (17.05.2002)

[Horvitz 99a]

Eric Horvitz. Uncertainty, Action, and Interaction: In Pursuit of Mixed-Initiative Computing. In: IEEE Intelligent Systems, September/October 1999, S. 17-20.

<ftp://ftp.research.microsoft.com/pub/ejh/mixedin.pdf> (12.09.2002)

[Horvitz 99b]

Eric Horvitz. Principles of Mixed-Initiative User Interfaces. In: The CHI is the limit: human factors in computing systems – CHI 99 conference proceedings. 15 - 20 May 1999, Pittsburgh, PA. Hrsg.: Marian H. Williams. ACM, New York, 1999, S. 159-166.

<ftp://ftp.research.microsoft.com/pub/ejh/chi99horvitz.pdf> (12.09.2002)

[Horvitz et al. 98]

Eric Horvitz, Jack Breese, David Heckerman, David Hovel, Koos Rommelse. The Lumière Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, July 1998. Morgan Kaufmann, San Francisco, 1998, S. 256-265. <ftp://ftp.research.microsoft.com/pub/ejh/lum.pdf>

(11.04.2002)

[Horvitz o.J.]

Eric Horvitz. Lumiere Project: Bayesian Reasoning for Automated Assistance. o.J.

<http://research.microsoft.com/~horvitz/lum.htm> (13.09.2002)

[Horvitz/Paek 99]

Eric Horvitz, Tim Paek. A Computational Architecture for Conversation. In: UM99 User Modeling: Proceedings of the Seventh International Conference, Banff, Canada, June 20 - 24, 1999. Springer, Wien, New York, 1999, S. 201-210. <ftp://ftp.research.microsoft.com/pub/ejh/converse.pdf>

(13.09.2002)

[Houchin 2001]

Dani Houchin. Building a learning and support strategy for Web-based applications. ITworld.com, 2001. <http://www.itworld.com/Man/3889/ITW010517houchin/> (24.04.2002)

[Houser/Deloach 98]

Rob Houser, Scott Deloach. Learning from Games: Seven Principles of Effective Design. In: Technical Communication, Vol. 45, No. 3. 1998, S. 319-329.

<http://www.techcomm-online.org/issues/v45n3/pdf/0277.pdf> (04.10.2002)

[ISO 9241-10]

Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 10: Grundsätze der Dialoggestaltung (ISO 9241-10 : 1996). Hrsg.: DIN Deutsches Institut für Normung e.V. Beuth, Berlin, 1996.

[ISO 9241-13]

Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 13: Benutzerführung (ISO 9241-13 : 1998). Hrsg.: DIN Deutsches Institut für Normung e.V. Beuth, Berlin, 1998.

[Johnson/Bellman/Lohse 2002]

Eric J. Johnson, Steven Bellman, Gerald L. Lohse. Cognitive Lock-In and the Power Law of Practice. Columbia Business School, Columbia University, 2002.

<http://www.cebiz.org/PDF%20Papers/power%20law%20submit.pdf> (23.04.2002)

[Kearsley 98]

Greg Kearsley. Minimalism: An Agenda for Research and Practice. In: Minimalism Beyond the Nurnberg Funnel. Hrsg.: John M. Carroll. MIT Press, Cambridge, Massachusetts u.a., 1998, S. 393-405.

[Krämer et al. 2002]

Nicole C. Krämer, Sabine Rüggenberg, Christoph Meyer zu Kniendorf, Gary Bente. Schnittstelle für alle? Möglichkeiten zur Anpassung anthropomorpher Interface Agenten an verschiedene Nutzergruppen. In: Mensch & Computer 2002: Vom interaktiven Werkzeug zu kooperativen Arbeits- und Lernwelten. Hrsg.: Michael Herzceg, Wolfgang Prinz, Horst Oberquelle. B. G. Teubner, Stuttgart u.a., 2002, S. 125-134.

[Krämer/Bente 2001]

Nicole C. Krämer, Gary Bente. Mehr als Usability: (Sozial-)psychologische Aspekte bei der Evaluation von anthropomorphen Interface-Agenten. In: i-com, Zeitschrift für interaktive und kooperative Medien, Heft 0-2001, S. 26-31.

[Krämer/Bente 2002]

Nicole C. Krämer, Gary Bente. Virtuelle Helfer: Embodied Conversational Agents in der Mensch-Computer-Interaktion. In: Virtuelle Realitäten. Hrsg.: Gary Bente, Nicole C. Krämer, A. Petersen. Hogrefe, Göttingen, 2002, S. 203-225. <http://www.embassi.de/publi/veroeffent/Kraemer-Bente.pdf> (30.08.2002)

[Krämer/Nitschke 2002]

Nicole C. Krämer, Julia Nitschke. Ausgabemodalitäten im Vergleich: Verändern sie das Eingabeverhalten der Benutzer? In: Bedienen und Verstehen. 4. Berliner Werkstatt Mensch-Maschine-Systeme. Hrsg.: R. Marzi, V. Karavezyris, H.-H. Erbe, K.-P. Timpe. VDI-Verlag, Düsseldorf, 2002, S. 231-248. Aus Mail von Autorin vom 09.09.2002

[Lassig 2001]

Marcel Lassig. Tests mit Anwendern. 2001, Lehrunterlagen. http://wwwswt.informatik.uni-rostock.de/deutsch/Mitarbeiter/michael/lehre/Usab_WS2000/Vortrag08/Usabvortrag.htm (24.04.2002)

[Lewis 98]

Michael Lewis. Designing for Human-Agent Interaction. In: AI Magazine, Vol. 19, No. 2. 1998, S. 67-78. <http://www2.sis.pitt.edu/~mlewis/aaaipap.ps> (06.09.2002)

[Lewis/Rieman 93]

Clayton Lewis, John Rieman. Task-Centered User Interface Design. 1993. <ftp://ftp.cs.colorado.edu/pub/distribs/clewis/HCI-Design-Book/> (11.04.2002)

[Lineback 2002]

Nathan Lineback. Microsoft Bob Version 1.00. 2002. <http://toastytech.com/guis/bob.html> (07.11.2002)

[Linton et al. 2000]

Frank Linton, Deborah Joy, Hans-Peter Schaefer, Andrew Charron. OWL: A Recommender System for Organization-Wide Learning. In: Educational Technology & Society, Vol. 3, No. 1. 2000, S. 62-76. http://ifets.ieee.org/periodical/vol_1_2000/linton.pdf (18.06.2002)

[Linton/Joy/Schaefer 99]

Frank Linton, Deborah Joy, Hans-Peter Schaefer. Building User and Expert Models by Long-Term Observation of Application Usage. In: UM99 User Modeling: Proceedings of the Seventh International Conference, Banff, Canada, June 20 - 24, 1999. Hrsg.: Judy Kay. Springer, Wien, New York, 1999, S. 129-138. <http://www.cs.usask.ca/UM99/Proc/linton.pdf> (11.04.2002)

[Marion 97]

Craig Marion. What is Performance-Centered Design? 1997. <http://www.chesco.com/~cmarion/PCD/WhatIsPCD.html> (09.08.2002)

[McGrenere/Moore 2000]

Joanna McGrenere, Gale Moore. Are We All In the Same "Bloat"? In: Proceedings Graphics Interface 2000, S. 187-196. <http://www.graphicsinterface.org/proceedings/2000/144/PDFpaper144.pdf> (19.06.2002)

[Mergel 98]

Brenda Mergel. Instructional Design & Learning Theory. 1998. <http://www.usask.ca/education/coursework/802papers/mergel/mergel.PDF> (02.08.2002)

[Microsoft 96]

IntelliSense in Microsoft Office 97. In: Microsoft Office 97 Whitepaper. Microsoft Corp., 1996. <http://www.ai.univie.ac.at/~paolo/lva/vu-sa/ps/o97iswp.color.ps.gz> (06.09.2002)

[Microsoft 2001]

Microsoft Schülermagazin. Microsoft Corp., 2001. <http://www.microsoft.com/germany/ms/schuelermagazin/reportagen/richtiglernen/lernen/mehrlinks.htm> (11.04.2002)

[Mirel 98]

Barbara Mirel. Minimalism for Complex Tasks. In: Minimalism Beyond the Nurnberg Funnel. Hrsg.: John M. Carroll. MIT Press, Cambridge, Massachusetts u.a., 1998, S. 179-218.

[Newman 99]

Michael Newman. Bob is dead; long live Bob. In: Post-Gazette News. 1999. <http://www.post-gazette.com/businessnews/19990523bob6.asp> (07.11.2002)

[Nielsen 93a]

Jakob Nielsen. Usability Engineering. Morgan Kaufmann, San Francisco u.a., 1993.

[Nielsen 93b]

Jakob Nielsen. Noncommand User Interfaces. 1993. <http://www.useit.com/papers/noncommand.html> (02.08.2002)

[Nielsen 2002]

Jakob Nielsen. Avoiding Commodity Status. In: Jakob Nielsen's Alertbox. 2002. <http://www.useit.com/alertbox/20020203.html> (11.04.2002)

[Oppermann 94]

Reinhard Oppermann. Individualisierung von Benutzungsschnittstellen. In: Einführung in die Software-Ergonomie – Gestaltung graphisch-interaktiver Systeme: Prinzipien, Werkzeuge, Lösungen. Hrsg.: Edmund Eberleh, Horst Oberquelle, Reinhard Oppermann. Walter de Gruyter, Berlin u.a., 1994, S. 235-269.

[Paul 93]

Hansjürgen Paul. Exploratives Agieren – Ein Beitrag zur ergonomischen Gestaltung interaktiver Systeme. Universität Bremen, Dissertation, 1993.

[Pierotti 98]

Deniese Pierotti. Heuristic Evaluation – A System Checklist. Xerox Corporation, 1998. http://www.stcsig.org/usability/resources/toolkit/he_cklst.doc (11.04.2002)

[Preim 99]

Bernhard Preim. Entwicklung interaktiver Systeme: Grundlagen, Fallbeispiele und innovative Anwendungsfelder. Springer, Berlin u.a., 1999.

[Raskin 94]

Jef Raskin. Intuitive equals familiar. In: Communications of the ACM. Vol. 37, No. 9. 1994, S. 17. <http://www.asktog.com/papers/raskintuit.html> (24.04.2002)

[Raskin 2000]

Jef Raskin. The Humane Interface: New Directions for Designing Interactive Systems. Addison Wesley, Reading, Massachusetts u.a., 2000.

[Redish 98]

Janice Redish. Minimalism in Technical Communication: Some Issues to Consider. In: Minimalism Beyond the Nurnberg Funnel. Hrsg.: John M. Carroll. MIT Press, Cambridge, Massachusetts u.a., 1998, S. 219-245.

[Reese Hedberg 98]

Sara R. Hedberg. Is AI going mainstream at last? A look inside Microsoft Research. In: IEEE Intelligent Systems, March/April 1998, S. 21-25. <ftp://ftp.research.microsoft.com/pub/ejh/x2xins.lo.pdf> (12.09.2002)

[Reeves/Nass 96]

Byron Reeves, Clifford Nass. The media equation: how people treat computers, television, and new media like real people and places. CSLI Publications, Stanford, California, 1996.

[Rheinfrank/Norman 96]

John Rheinfrank, Don Norman. A Conversation with Don Norman. In: interactions online, Vol. 2, No. 2. 1996. <http://www.acm.org/interactions/vol2no2/interview/dnorman.htm> (07.11.2002)

[Richter/Naumann/Horz 2001]

Tobias Richter, Johannes Naumann, Holger Horz. Computer Literacy, computerbezogene Einstellungen und Computernutzung bei männlichen und weiblichen Studierenden. In: Mensch & Computer 2001: 1. Fachübergreifende Konferenz. Hrsg.: Horst Oberquelle, Reinhard Oppermann, Jürgen Krause. B. G. Teubner, Stuttgart, 2001, S. 71-80. <http://mc.informatik.uni-hamburg.de/konferenzbaende/mc2001/V23.pdf> (24.05.2003)

[Rosson/Carroll 2002]

Mary B. Rosson, John M. Carroll. Usability Engineering, Scenario-Based Development of Human-Computer Interaction. Morgan Kaufmann, San Francisco, 2002.

[Rühl 99]

Stefanie Rühl. Ängste und deren Überwindung durch Motivation beim Erlernen von Computeranwendungen. Freie Universität Berlin, Fachbereich Erziehungswissenschaft und Psychologie, 1999. <http://home.t-online.de/home/stefanie.ruehl/teil2.htm> (17.05.2002)

[Rupp 2001]

Katja Rupp. Microsoft übt sich in Selbstironie. In: TELEPOLIS – magazin der netzkultur. 2001. <http://www.ct.heise.de/tp/deutsch/inhalt/co/7387/1.html> (31.10.2002)

[Schaumburg 2001]

Heike Schaumburg. Computers as tools or as social actors? – User perspectives on the social interface. 2001. <http://userpage.fu-berlin.de/%7Ehoige/papers/oastudy1.html> (06.09.2002)

[Schudnagis/Womser-Hacker 2002]

Monika Schudnagis, Christa Womser-Hacker. Multimediale Lernsysteme softwareergonomisch gestalten: das Projekt SELIM. In: Mensch & Computer 2002: Vom interaktiven Werkzeug zu kooperativen Arbeits- und Lernwelten. Hrsg.: Michael Herzceg, Wolfgang Prinz, Horst Oberquelle. B. G. Teubner, Stuttgart u.a., 2002, S. 215-224.

[Sebrechts/Marsh/Furstenburg 90]

Marc M. Sebrechts, Richard L. Marsh, Charlotte T. Furstenburg. Integrative Modeling: Changes in Mental Models During Learning. In: Cognition, Computing, and Cooperation. Hrsg.: Scott P. Robertson, Wayne Zachary, John B. Black. Ablex, Norwood, New Jersey, USA, 1990, S. 338-398.

[Selber/Johnson-Eilola/Mehlenbacher 97]

Stuart A. Selber, Johndan Johnson-Eilola, Brad Mehlenbacher. Online Support Systems: Tutorials, Documentation, and Help. In: The Computer Science and Engineering Handbook. Hrsg.: A. B. Tucker, Jr. CRC Press, Boca Raton, Florida, 1997, S. 1619-1643.
http://www4.ncsu.edu/~brad_m/Research/Support97.pdf (04.06.2002)

[TBS 97]

Ergonomie-Prüfer – Handlungshilfe zur ergonomischen Gestaltung der Bildschirmarbeit. Technologieberatungsstelle beim DGB Landesbezirk NRW e.V., Oberhausen, 1997.

[Tidwell 99]

Jenifer Tidwell. Common Ground: A Pattern Language for Human-Computer Interface Design. 1999.
http://www.mit.edu/~jtidwell/common_ground_onefile.html (17.05.2002)

[Tognazzini 98]

Bruce Tognazzini. Intuitive vs. Familiar. 1998.
<http://www.asktog.com/columns/006intuitivsfamiliar.html> (24.04.2002)

[Tomasi/Mehlenbacher 99]

Martin D. Tomasi, Brad Mehlenbacher. Re-engineering Online Documentation: Designing Examples-based Online Support Systems. In: Technical Communication, Vol. 46, No. 1. 1999, S. 55-66.
http://www4.ncsu.edu/~brad_m/Research/Tomasi99.pdf (04.06.2002)

[UVV_VBG_104 95]

Unfallverhütungsvorschrift "Arbeit an Bildschirmgeräten", VBG 104. 1995.
<http://selab24.informatik.uni-bremen.de/Projekte/SET/setup/vbg10436.html> (26.04.2002)

[Van der Meij/Carroll 98]

Hans van der Meij, John M. Carroll. Principles and Heuristics for Designing Minimalist Instruction. In: Minimalism Beyond the Nurnberg Funnel. Hrsg.: John M. Carroll. MIT Press, Cambridge, Massachusetts u.a., 1998, S. 19-53.

[Wandke 2002]

Hartmut Wandke. Assistance in Human-Machine-Interaction – Conceptual Framework and a Proposal for Taxonomy. Submitted to: Theoretical Issues in Ergonomics Science. Deutsches Manuskript aus Mail von Nicole Krämer vom 18.09.2002.

[Wen 2000]

Hui-Fang Wen. Empirical Studies of Online Help. 2000. <http://www.otal.umd.edu/UUGuide/hui/> (04.06.2002)

[Winograd 96]

Terry Winograd. Microsoft Bob. In: Bringing Design to Software. Hrsg.: Terry Winograd, John Bennett, Laura De Young, Bradley Hartfield. ACM Press, New York, 1996, S. 146-150.

[Yom/Wilhelm 2002]

Miriam Yom, Thorsten H. Wilhelm. Zur Wirkung von Hilfefunktionen auf das Online-Shoppingerleben und Shoppingverhalten bei Web-Novizen. In: i-com, Zeitschrift für interaktive und kooperative Medien, Heft 2-2002, S. 18-23.