

Miß- und Nichtverstehen als Ursachen mißlingender Mensch-Computer-Interaktion

Der Computer als *cognitive load* beim Schreiben¹

Jörg Wagner
Halle

Text production processes – and this is true not only in the sciences – are highly complex by nature and determined by a large set of external (situational) and internal (cognitive) constraints. The writer's tool is mostly considered to belong only to the first of these. Word processors, though, while often advertised as being easy to use and self-explaining, have typically become ever more complex in their functionality. That is why interacting with the interface of such a program is often a heavy cognitive load that may distract the user from his/her actual work (i. e. to write a text). Computer novices and experts alike must often spend “non-task” time and attention using this complex software. One reason for this inefficiency is that the user does not understand or misunderstands verbal elements of the interface (e. g. menus, ‘dialogues’, error messages, etc.). This paper attempts to call the reader's attention to the role of the computer as a writing tool that is still far from perfect. Examples of semantic, grammatical, and pragmatic reasons for the occurrence of mis- and non-understandings that disrupt the writing process will be presented.

1 Einleitung

Der Computer hat sich in den Wissenschaften als Schreib-Gerät etabliert. Dies hängt fraglos auch damit zusammen, daß sich die Bedienungsfreundlichkeit in den letzten Jahren stark verbessert hat.

Dennoch verlangt die Interaktion mit dem *Interface* der verwendeten Software oft volle Konzentration (Kapitel 2). Fehlbedienungen sind selbst für erfahrene Nutzer keine Seltenheit. Unsere Hypothese lautet: Eine Quelle mißlingender Mensch-Computer-Interaktion sind die verbalen Interface-Elemente, die für den Nutzer miß- oder unverständlich sein können (Kapitel 3 und 4). Um dies zu zeigen, wurde eine kontrollierte Interaktionsstudie mit Studierenden durchgeführt (Kapitel 5). Die darin gewonnenen Daten mißlingender Mensch-Computer-Interaktion wurden analysiert und durch Beispiele aus realen Arbeitsprozessen im universitären Bereich ergänzt (Kapitel 6). Die Ergebnisse zeigen, daß die Angewandte Linguistik

1 Ich danke allen, die mit Kritik und Anregungen den vorliegenden Text in entscheidendem Maße geprägt haben, insbesondere den Herausgeberinnen dieses Bandes, Dagmar Knorr und Eva-Maria Jakobs, sowie Rolf Todesco.

einen Beitrag zur Verfeinerung der Methoden des *usability engineering* zu leisten in der Lage sein sollte (Kapitel 7).

2 Kognitive *Constraints* bei der schriftlichen Textproduktion

Schriftliche Textproduktion – und an dieser Stelle sei neben der wissenschaftlichen zumindest auch die schulische erwähnt – ist ein hochkomplexer Prozeß, der von einer Vielzahl situativer bzw. externer sowie individueller Faktoren bestimmt wird (vgl. stellvertretend Antos 1989; Krings 1992; Jakobs 1995). Zu ersteren werden in der Literatur vor allem der *Ort* und der *zeitliche Rahmen*, der *institutionelle Kontext* und das *Arbeitsmedium*, aber auch der *bereits produzierte (Teil-)Text* gezählt.

Innerhalb der individuellen Faktoren – zu denen z. B. (fachliches und sprachliches) *Wissen*, *Motivation* und *körperliche Verfassung* gehören – spielt neben der *Schreibverfahren* vor allem die individuelle *kognitive Belastbarkeit* eine Rolle. Parallelität, Interaktivität und Rekursivität von Planungs-, Formulierungs- und Revisionsprozessen stellen den Schreibenden vor höchste Anforderungen: “Text production, especially writing, seems to operate near the threshold of overload ...” (de Beaugrande 1984, 96).

Durch die „Verteilung der Aufmerksamkeit auf die verschiedenen Prozesse während der Textproduktion“ kann es zu „Engpässe[n] bei der kognitiven Verarbeitung“ kommen (Molitor-Lübbert 1989, 283): De Beaugrande (1984, 96) prägte dafür den anschaulichen Begriff des „Flaschenhalses“ (*bottleneck*). Der Schreibende muß – da er nicht in der Lage ist, allen Anforderungen des Schreibprozesses gleichzeitig zu genügen – seine „Aufmerksamkeit selektiv und schwerpunktmäßig abwechselnd auf die eine oder andere Prozeßebene“ richten. Kleinste Ablenkungen oder zusätzliche Probleme können dieses ständige Wechselspiel zwischen der „Belastung des sensorischen Kurzzeitgedächtnisses, des Arbeitsgedächtnisses, des Kurzzeitgedächtnisses und des Langzeitgedächtnisses“ (Molitor-Lübbert 1989, 283) aus dem Gleichgewicht bringen.

Um so erstaunlicher erscheint vor diesem Hintergrund die Tatsache, daß immer mehr Schreiberinnen und Schreiber in den Wissenschaften mit dem Werkzeug *Computer* arbeiten (vgl. Knorr 1995). Der relativ hohe Geräuschpegel, die den Schreibfluß unterbrechenden automatischen Systemabläufe (wie z. B. periodische Speicherungen und selbst solche zunächst trivial erscheinenden Dinge wie *Bildschirmshoner*) oder auch die Unwägbarkeiten von Systemabstürzen scheinen jedoch von den individuellen Vorteilen des Computereinsatzes mehr als wettgemacht zu werden. Selbst als störend empfundene Systemeigenschaften werden, wenn auch klagend, hingenommen. So berichtete eine Probandin der durchgeführten Studie, zu ihren Erfahrungen mit verschiedenen Textverarbeitungssystemen befragt, daß die automatische Rechtschreibkorrektur bzw. die Markierung von Fehlern (in *Word für Windows 6.0TM*) sie ablenken würden. Wie man diese (standardmäßig aktivierten) Optionen deaktivieren kann, war ihr jedoch nicht bekannt.

Vor allem unter kognitiven Aspekten scheint der Computer m. E. unterschätzt: Der mit Hilfe eines Textverarbeitungssystems Schreibende hat es aber – insofern er den Computer nicht lediglich als Schreibmaschine zum „Abtippen“ (fast) fertiger Manuskripte nutzt (eine oft geringgeschätzte, aber doch wohl völlig legitime Verwendungsweise des Computers) – mit einem äußerst komplexen Werkzeug zu tun. Dieses kann sich als zusätzliches kognitives *Constraint* bei der Textproduktion bemerkbar machen: *Papier und Bleistift* sind fraglos mit geringerem kognitiven Aufwand einzusetzen. Die Aufmerksamkeit – ohnehin zwischen den Schreibprozeßstufen verteilt – muß nun auch noch den Interaktionsprozessen mit dem Computer gewidmet werden. Im Extremfall verlangt das Werkzeug mehr Konzentration als der zu schreibende Text, und dann kommt nicht nur der

Perfektionist ... vor lauter Handbuchwälzen, Druckformatvorlagen-Definieren, automatischer Gliederungsnumerierung, Index- und Makro-Erstellung etc. nicht mehr zum Denken, Redigieren, Formulieren und Reformulieren (Schmitz 1995, 160).

Der Einfluß des Computers auf Textproduktionsprozesse (d. h. eventuelle Veränderungen globaler oder lokaler Planungs-, Formulierungs- und Revisionsvorgänge) und insbesondere auf die Textqualität ist nach wie vor weitgehend ungeklärt. In Gesprächen wird jedoch immer häufiger der (subjektive) Eindruck kolportiert, daß Texte in ihrem Erscheinungsbild zwar vollkommener würden, inhaltlich und/oder stilistisch aber zunehmend zu wünschen übrig ließen. Schmitz' Beobachtungen an einem konkreten Textbeispiel (1995, 154ff.) scheinen diesen Eindruck zu bestätigen.

Diese Entwicklung – so sie tatsächlich nachzuweisen sein sollte – wird im Kontext wissenschaftlicher Publikationen durch die zunehmende Verlagerung der gestalterischen Arbeit von den Verlagen zu den Autoren (Jakobs/Knorr 1995) noch verstärkt. Die traditionelle personelle Trennung zwischen den Fach-Experten und den Experten für die Präsentation wird damit aufgehoben. Autoren (insbesondere jüngere) sind dadurch *gezwungen*, sich sehr intensiv mit Fragen der Präsentation und deren technischer Realisierung zu befassen.

Ähnlich gelagerte Probleme lassen sich in den Bereichen der Schule oder der universitären Ausbildung finden. Die Fähigkeit zur Nutzung moderner Medien tritt in ihrer Bedeutung immer mehr neben die traditionellen Kulturtechniken *Lesen*, *Schreiben* und *Rechnen*. Das drückt sich auch darin aus, daß Kurse zur Einführung in die elektronische Textverarbeitung an vielen Schulen inzwischen zum verbindlichen Kanon zählen (vgl. Rahmenrichtlinien Deutsch für Sekundarschulen in Sachsen-Anhalt). Daß sich dadurch die Leistungen der Schülerinnen und Schüler beim Produzieren von Texten verbessert hätten – ein oft postuliertes Ziel (vgl. Plieninger 1991, 23; Wagner 1995) – konnte bisher kaum nachdrücklich nachgewiesen werden; lediglich die durch die Computernutzung erhöhte Motivation von eher schwächeren Schülern scheint zu besseren Resultaten zu führen (vgl. Bangert-Drowns 1993, 87f.).

An den Hochschulen und Universitäten etabliert sich inzwischen ein neuer Quasi-Standard: Selbst Seminararbeit werden zunehmend unter Nutzung von Computern verfaßt. Ob dabei die eigene Entscheidung der Studierenden für das Werkzeug *Computer* den Ausschlag gibt oder der Zwang, hinsichtlich formaler Aspekte nicht hinter vermeintlichen (weil oft nicht einmal eingeforderten) Standards zurückbleiben zu dürfen, ist in vielen Fällen sicher kaum zu entscheiden.

Viele Studentinnen und Studenten – gerade in den Geistes- und Sozialwissenschaften – finden erfahrungsgemäß oft erst dann zum Computer als Schreibwerkzeug, wenn es eine längere Arbeit (etwa eine Abschlußarbeit) zu bewältigen gilt. Die unbestrittenen Vorteile des Mediums gegenüber *Papier und Bleistift* (insbesondere die leichte Überarbeitbarkeit von bereits Geschriebenem) geben dann den Ausschlag. Dies ist aber m. E. der ungünstigste Zeitpunkt, um sich mit den Unwägbarkeiten von Computertechnik zu befassen. Notwendigerweise wird – zumindest zu Beginn – ein hoher Lernaufwand abverlangt. Außerdem braucht der Computer als Schreibwerkzeug Gewöhnung. Gerade in dieser Phase des Studiums erscheint es jedoch als angeraten, kognitive Belastungen, die nicht direkt zum Prozeß der Textproduktion zählen, nach Möglichkeit zu minimieren.

Ebenso sollten Anleitungen zum (wissenschaftlichen) Schreiben nicht unbedingt im Zusammenhang mit Computern gedacht werden. Eigene Erfahrungen haben gezeigt, daß technischen Erläuterungen und Übungen während der Seminare Tribut zu zollen ist; für das eigentliche Schreiben-Lernen bleibt weniger Zeit und Aufmerksamkeit. Empfehlungen, die es nahelegen, Studentinnen und Studenten alle Möglichkeiten eines Textverarbeitungsprogrammes nahezubringen (van Waes 1994, 56), um zu verhindern, daß sie den Computer als eine bessere Schreibmaschine benutzen, gehen deshalb vielleicht in die falsche Richtung.

Ein Ziel des vorliegenden Textes ist es darum, den Computer einmal nicht aus wohlwollender Perspektive als ein das Schreiben unterstützendes Werkzeug zu betrachten, sondern seine Komplexität und Problematik ins Bewußtsein zurückzurufen. Von einem „happy medium“, welches „effortless creativity“ (Schmitz 1995, 149) beim Schreiben unterstützt, sind heutige kommerzielle Hard- und Software jedenfalls noch weit entfernt.

Für das (wissenschaftliche) Schreiben sind jedoch *wirklich* benutzerfreundliche Systeme notwendig, welche es den Schreibenden erlauben, sich gedanklich voll auf die Arbeit zu konzentrieren. So wie Umberto Eco es in seinem Roman „Das Foucaultsche Pendel“ dem Lektor Jacopo Belbo in die Computer-Tastatur diktiert:

Der Computer hilft einem nicht denken, aber er hilft einem für ihn denken. Total spirituelle Maschine. [...] ... mit ihm (ihr?) phantasieren die Finger, der Geist streift die Tasten, die Gedanken fliegen auf goldenen Schwingen, endlich meditiert die strenge Kritische Vernunft über das Glück des ersten Anhiebs (Eco 1988, 35f.).

3 Mensch-Computer-Interaktion

Neben den *Interaktionen mit dem* entstehenden *Text* sowie der *Umwelt* befindet sich der Benutzer von Textverarbeitungssystemen in einer *Interaktion mit dem System* aus Hard- und Software. Unter *Interaktion* wird in der vorliegenden Arbeit der wechselseitige Austausch von *Daten* zwischen Nutzer und System verstanden: *Daten*, die das System dem Benutzer über den visuellen oder auditiven Kanal zur Verfügung stellt, werden im Prozeß der Verarbeitung durch den Benutzer zu *Informationen* und können beim ihr/ihm bestimmte Handlungen evozieren. Die Eingaben des Benutzers wiederum führen zu Zustandsänderungen des Systems (vgl. Sagawe 1989, 297; zur Unterscheidung zwischen *Daten* und *Informationen* vgl. Siefkes 1989, 333).

Die Interaktion zwischen Benutzer und Computer wird stark von kognitionspsychologischen Aspekten bestimmt. Wahrnehmungsselektion, Aufmerksamkeitssteuerung, Gedächtnisleistungen, Prozesse der Sprach- und Informationsverarbeitung, Problemlöseprozesse oder das Fällen von Entscheidungen bestimmen in entscheidendem Maße über das Gelingen oder Mißlingen der Interaktion. Dabei wird der Informationsfluß zwischen Benutzer und Computer über das Interface zum entscheidenden Kriterium:

One of the most important features ... is the flow and feedback of information through the interface. The user needs information from the computer and the computer cannot function without information from the user. [...] Smooth operation of the system requires a timely flow of information that is relatively free of error states in the machine and in the user (Norman 1991, 12).

Fehlerzustände des Systems sind gut definiert: Eingaben des Nutzers können außerhalb des vorprogrammierten Rahmens liegen oder nicht mit der gerade erforderlichen Art der Eingabe übereinstimmen. Unzureichende Systemressourcen oder Befehle bzw. Verweise auf nicht existierende Funktionen zählen ebenfalls dazu (vgl. Norman 1991, 12f.).

Fehler des Benutzers sind dagegen sehr schwer zu definieren. Sie erwachsen oft aus der Ratlosigkeit des Benutzers, was mit einer Mitteilung oder Aufforderung des Computers anzufangen sei, d. h. welche Handlung vom System erwartet wird.

Mensch-Computer-Interaktion ist ein semiotischer Prozeß. Der Nutzer hat es in diesem vor allem mit arbiträren Zeichen zu tun. Arbitrarität zeichnet selbst die meisten der Verwendung findenden „piktogramm-haften“ Zeichen (*'icons'*) aus (wie sie bspw. in Menüleisten vorkommen). Im Gegensatz zu echten Piktogrammen, die sich zu den von ihnen repräsentierten Objekten in einem Ähnlichkeitsverhältnis befinden und oft weltweit verständlich sind, kann die Bedeutung der meisten dieser graphischen Zeichen nicht ohne Vorwissen erschlossen werden.

Ein hoher Anteil des Informationsflusses vom Computer zum Benutzer wird jedoch über die verbalen Elemente der Benutzerschnittstellen realisiert. Zu den verbalen Interface-Elementen graphischer Nutzeroberflächen eines typischen Textverarbeitungssystems zählen: Programmnamen; Dateinamen; Dateinamen-Exten-

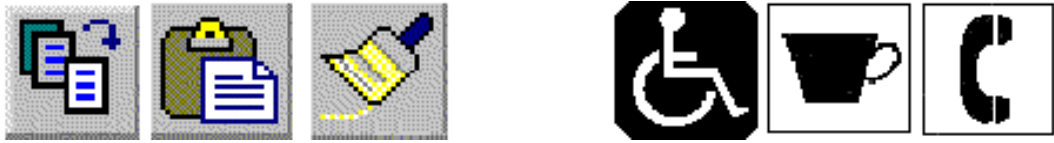


Abb. 1: Icons und Piktogramme

sionen; Menüleisten; Menübefehle; Dialogfenster; Angaben zu Formatvorlagen, Schriftart und Schriftgröße; Angaben in der Statuszeile (Seite, Abbildungen, Zeile, Spalte); Quickinfo bzw. *ToolTips*; Hilfesysteme; „Tips und Tricks“ sowie Fehlermeldungen und Hinweise.

4 Mißlingende Interaktion: Miß- bzw. Nichtverstehen

Mißlingende Interaktion zwischen Mensch und Computer kann die verschiedensten Ursachen haben. Neben Wahrnehmungsproblemen, Orientierungsverlust, mangelnder Aufmerksamkeit oder Motivation des Benutzers, die hier nicht Gegenstand sein sollen, kann das Miß- bzw. Nichtverstehen verbaler Interface-Elemente durch den Benutzer Ursache mißlingender Interaktionsprozesse sein.

In der Mensch-Mensch-Kommunikation sind Mißverständnisse bzw. deren Ursachen z. T. schwer zu lokalisierende Phänomene; sie können den beteiligten Kommunikanten sogar verborgen bleiben (vgl. Blum-Kulka/Weizman 1988). Oft besteht auch gar nicht der Anspruch, alles bis ins Detail zu verstehen: Eine Idee beispielsweise, die aus einem anspruchsvollen Aufsatz entnommen werden kann, wird dann subjektiv höher bewertet als das Miß- oder Nichtverstehen des überwiegenden Teils des Textes.

Anders verhält es sich bei der Mensch-Computer-Interaktion: Das korrekte und weitestgehend vollständige Verstehen ist Voraussetzung für eine erfolgreiche Interaktion. Bedeutungen sind auf seiten des Computers klar definiert: Es besteht eine eindeutige Zuordnung von Interface-Elementen zu Programmabläufen. Wenn diese – hier interessierenden verbalen – Interface-Elemente vom Benutzer nicht oder mißverstanden werden, kommt es in den meisten Fällen zu einer Fehlbedienung des Systems. Miß- bzw. Nichtverstehensleistungen des Systembenutzers lassen sich deshalb relativ leicht an einem rein pragmatischen Kriterium festmachen: dem Erreichen bzw. Nicht-Erreichen des jeweiligen konkreten Handlungsziels.

Allein die Tatsache, daß die „Bedeutungen vorgegeben sind“, führt jedoch nicht, wie Peters (1990, 73) behauptet, dazu, daß es in der Mensch-Computer-Interaktion „sogar weniger potentielle ‚Mißverständnisse‘“ als in der Mensch-Mensch-Kommunikation gibt. Gerade die völlige Bestimmtheit in der Bedeutung erlaubt keinen Spielraum beim Verstehen. Menschliche Kommunikation funktioniert oft gerade wegen ihrer Vagheit. Und: Falls Bedeutungen unklar bleiben oder divergieren, kön-

nen diese – im Gegensatz zur Mensch-Computer-Interaktion – zwischen den Kommunikationspartnern ausgehandelt werden.

5 Design der Interaktionsstudie

An dieser Stelle soll der Versuch unternommen werden, Ursachen für Miß- bzw. Nichtverstehen als Auslöser mißlingender Mensch-Computer-Interaktion aufzuzeigen. Das der Analyse zugrundeliegende Korpus besteht aus Daten, die in einer kontrollierten Interaktionsstudie gewonnen wurden, ergänzt durch Erfahrungen, die von mir sowie Kolleginnen und Kollegen beim täglichen Umgang mit Computern gemacht bzw. berichtet wurden.

Die durchgeführte Studie lehnt sich im Design eng an Methoden gängiger *usability tests* (Test von Software hinsichtlich ihrer Benutzerfreundlichkeit) an. Norman (1991, 19ff.) unterscheidet die folgenden drei grundlegenden methodischen Varianten der empirischen Software-Ergonomie-Forschung:

- (1) Beobachtung der Benutzer in der Interaktion mit dem System (unter Einbeziehung von Videoaufnahmen und/oder *thinking-aloud-protocols*)
- (2) Befragungen von Computer-Nutzern (meist mittels standardisierter Fragebögen) zur Erfassung subjektiver Faktoren der Mensch-Computer-Interaktion
- (3) Experimente (zur Bestimmung des Effekts einzelner Design-Faktoren).

Um die Hypothese, wonach mißlingende Mensch-Computer-Interaktion zu einem großen Teil auf Miß- bzw. Nichtverstehen verbaler Interface-Elemente durch den Benutzer zurückgeführt werden kann, zu überprüfen bzw. zu festigen, wurde für die durchgeführte Studie Methode (1), die Beobachtung, ausgewählt.

Es wurden fünf Versuchsgruppen zu jeweils zwei Studentinnen bzw. Studenten mit einer Aufgabe konfrontiert, die den Umgang mit dem Textverarbeitungsprogramm *Word für Windows* (Version 2.0) beinhaltete. Die Aufgabe war so angelegt, daß sie nicht über Routinearbeiten (etwa einer studentischen Hilfskraft, die mit der Überarbeitung eines Manuskriptes beauftragt ist) im universitären Alltag hinausging. Sie bestand darin, einen auf Diskette vorliegenden Text nach bestimmten Vorgaben zu formatieren sowie ein ebenfalls auf der Diskette befindliches Bild an einer markierten Stelle im Text einzufügen.

Während der Arbeit wurde der Bildschirminhalt auf Videoband mitgeschnitten. Dies wurde den Probanden mitgeteilt. Nicht mitgeteilt wurde dagegen, daß auch der Dialog, den die Probanden während der Arbeit führten, aufgezeichnet wurde. Auf diese Weise konnten Daten sowohl aus der *Interaktion mit dem System* als auch aus der *Kommunikation über das System* gewonnen werden.

Die Videodaten gaben Aufschluß über kritische Momente in der Interaktion der Probanden mit dem Interface. An diesen Stellen wurde der Dialog der Probanden transkribiert und dazu herangezogen, Ursachen für die Interaktionsprobleme aufzudecken.

Die Beispiele auf den folgenden Seiten beziehen sich – insofern nicht anders gekennzeichnet – auf den Umgang mit dem Textverarbeitungssystem *Word für Windows 2.0™*. Daß die Software-Entwicklung nicht an dieser Stelle stehengeblieben ist und einige Probleme in neueren Versionen dieses Programms gelöst sind, ändert m. E. nichts an der Aussagekraft der Beispiele in unserem Kontext. Außerdem treten, wie zu zeigen sein wird, in neuen Programmversionen auch neue Probleme auf. Es wird im folgenden ausschließlich auf Mängel sehr intensiv gepflegter Applikationen großer Software-Firmen hingewiesen. Eine Vielzahl der im universitären oder auch im Büro-Alltag eingesetzten Software wird jedoch von kleineren Firmen oder Privatpersonen entwickelt, die naturgemäß noch viel weniger Sorgfalt auf das Design ihrer Produkte verwenden (können).

Zu klären bleibt, welche der im folgenden diskutierten Mißverständnisquellen von einer mangelhaften Übersetzung der Software aus dem Englischen herrühren. Erste Vergleiche deuten jedoch darauf hin, daß die Unterschiede zwischen englischer und deutscher Fassung eher marginaler Natur sind, wenngleich einige Mißverständnispotentiale tatsächlich erst durch die Übersetzung entstehen.

6 Ursachen von Miß- und Nichtverstehensprozessen

Ursachen für Miß- und Nichtverstehensprozesse konnten in den Bereichen der Semantik, der Grammatik wie auch der Pragmatik aufgedeckt werden. Eine Wichtung der Ursachen (d. h. auch Angaben über deren Häufigkeit bzw. Schwere) muß umfangreicheren Studien sowie Experimenten zu konkreten Design-Faktoren vorbehalten bleiben.

6.1 Semantische Ursachen für Miß- bzw. Nichtverstehen

6.1.1 Bedeutungsinterferenzen

In der Interaktion zwischen Mensch und Computer stehen sich bezogen auf ein und dasselbe *Signifiant* unterschiedliche *Signifiés* gegenüber: Softwareseitig wird beispielsweise von einem Menübefehl eine eindeutig zugeordnete Kette von Operationen (ein festgelegter Programmablauf) repräsentiert, während der Begriff für den Benutzer, vor allem den Novizen, mit den Bedeutungen aus dessen Welt- und Alltagswissen belegt ist.

Beispiel: In dem Befehl *Anmerkung* im Menü *Einfügen* entdeckt ein unerfahrener Benutzer wahrscheinlich die Möglichkeit zum Einfügen einer Anmerkung im Sinne einer *Endnote*. Tatsächlich wird jedoch ein Eingabefeld geöffnet, in welchem ein Überarbeitungsvermerk (also beim Ausdruck nicht erscheinender Text) eingegeben werden kann.

6.1.2 Inkonsistenz in der Bezeichnung

Einmal für bestimmte Programmabläufe gewählte Bezeichnungen sollten konsistent gebraucht werden. Ein Benutzer, der sich in die Bezeichnungen einer Anwen-

dung eingearbeitet hat, wird dadurch in die Lage versetzt, durch Transfer dieses Wissens auch ähnliche Applikationen bedienen zu können.

Wie fatal sich Veränderungen in der Bezeichnung von Befehlen auswirken können, zeigte sich auch in der Studie: Eine Probandin wollte das von der Aufgabenstellung geforderte Einrücken der ersten Zeile eines jeden Absatzes elegant mit Hilfe der sogenannten *Druckformatvorlagen* bewerkstelligen. Ihr war dafür der Befehl *Formatvorlage...* im Menü *Format* geläufig. Im zu verwendenden Programm heißt diese Funktion jedoch *Druckformate...* Allein diese relativ geringe Abweichung genügte, um die Interaktion mißlingen zu lassen.

6.1.3 Polysemie

Eng zusammenhängend mit dem Problem unterschiedlicher Referenzen muß das Problem der Polysemie betrachtet werden. Sprachliche Zeichen werden – oft sogar innerhalb eines Programms – für verschiedene Funktionen verwendet. Ein Beispiel dafür ist die Verwendung von *Index*: Im Menü *Einfügen* bedeutet *Index*, daß für den in Bearbeitung befindlichen Text ein solcher erstellt bzw. eingefügt wird, während nach Aufruf des Befehls *Index* im Menü *Hilfe* eine Liste von Themen der Online-Hilfe angezeigt wird.

Auch das Signifiant *Einfügen* ist mehrfach besetzt: Man findet es unter dem Menüpunkt *Bearbeiten* als *Inhalte einfügen* (Was eigentlich sind ‚Inhalte‘?), als Menüpunkt *Einfügen* sowie im Menü *Tabelle* (*Tabelle einfügen*). Warum alle diese Programmoptionen nicht in einem Menü zusammengefaßt werden, ist schwer einzusehen, werden doch durch diese Verteilung Suchprozesse verlängert und Mißverständnisse provoziert.

Als sehr problematisch hat sich in der Interaktionsstudie die Polysemie von *Datei* erwiesen. *Datei* hat mindestens zwei Bedeutungen, eine weitere (*Datei₁*) und eine engere (*Datei₂*):

Datei₁ = jede mit einem Namen versehene Software-Entität (Programmdateien, Systemdateien, „Daten-Dateien“: Grafiken, Texte, Datenbanken etc.)

Datei₂ = ausschließlich TEXT-Dateien

Dies führte bei einer der Versuchsgruppen sogar zum Systemabsturz, als nämlich versucht wurde, die auf Diskette befindliche Grafik als *Datei* (was sie ja zweifellos im Sinne der ersten Bedeutung ist) in den Text zu laden. Im Menü *Einfügen* ist jedoch *Datei* nur im eingeschränkten Sinne, nämlich als Text-Datei zu verstehen.

6.1.4 Kategorisierung und Hierarchisierung von Informationen

Neben der Entscheidung des System-Entwicklers für einen konkreten Zeichenträger beeinflußt auch die Anordnung dieser Zeichen in systematischen Hierarchien die Qualität des Verstehens des Benutzers. Sinnvolle Begriffshierarchien können den Benutzer kognitiv stark entlasten, Suchprozesse minimieren und Mißverstehen reduzieren. An dieser Stelle kann der Begriff der *family resemblance*, wie er von Rosch und Mervis (1975, 576) geprägt wurde, als Erklärungsmuster dienen: die

prototypischen Elemente (hier: Befehle) einer Kategorie (hier: eines Menüs) sind diejenigen, die die meisten Eigenschaften mit anderen Elementen dieser Kategorie gemein haben.

Probleme ergeben sich, wenn z. B. der Befehl *Seite einrichten*, den man in früheren Versionen von *Word für Windows* (sinnvollerweise?) im Menü *Format* zu finden war, in der Version 6.0 unter *Datei* eingeordnet ist.

6.1.5 Probleme der Deixis

Miß- bzw. Nichtverstehen wird häufig dadurch provoziert, daß deiktische Elemente ganz fehlen oder dem Benutzer wichtige Komponenten der „Sprech“situation, auf die verwiesen wird, unbekannt sind. Die Einbindung von „Äußerungen“ (Fehlermeldungen, Warnungen, Hilfe-Texten) in die konkrete Anwendungs- (d. h. „Sprech-“) Situation wird dem Benutzer erschwert, wenn nicht unmöglich gemacht.

Beispiel: Bei der Meldung „Überprüfe Dateisystem“ (etwa bei der Installation neuer Software) können nur außersprachliche Anzeichen weiterhelfen (z. B. Geräusche des Systems, die darauf schließen lassen, daß die fragliche Aktion gerade ausgeführt wird). Die *personale Deixis* ist zweideutig, so daß dem Benutzer möglicherweise unklar ist, ob er als Adressat (eines Imperativs mit Ellipse des Subjekts) zur fraglichen Operation aufgefordert wird oder ob lediglich eine Mitteilung des Systems (in der Ich-Origo) über sich selbst vorliegt. (Hier ist der Fehler wohl in der Übersetzung aus dem Englischen zu suchen: Die formal gut zu unterscheidenden grammatischen Formen „Checking...“ bzw. „Please, check...“ werden im Deutschen zu „[ich] überprüfe [gerade]“ bzw. „Überprüfe [du]!“.)

6.1.6 Metaphorik und Analogiebildung

Der Gebrauch von Metaphern im Umfeld von Technik und insbesondere von Computern ist schon mehrfach Gegenstand extensiver Betrachtungen gewesen. Weingarten (1989, 86) hat gezeigt, „wie Technik als Sprache verstanden wird“ und wie dabei Metaphern und Analogien auf intensive Weise dazu beitragen, neue Gegenstände begrifflich zu erschließen.

Von vielen System-Entwicklern und Software-Ergonomen wird die Etablierung von „Benutzerillusionen“ als äußerst wichtig eingeschätzt: “‘METAPHOR’ ... seems to be the holy grail at Apple.” (Erickson 1990, 65).

Dabei entscheiden oft die Vorstellungsbilder, „die sich die Softwareentwickler ... von ihrem Gegenstand machen“ (Weingarten 1989, 98), über die Wahl einer Metapher. An der Software-Entwicklung beteiligte Experten „leben“ und interagieren sprachlich jedoch zwangsläufig in einer anderen Diskurswelt als die meisten Endnutzer der Systeme (vgl. dazu auch Floyd 1990). Gewählte Metaphern basieren dadurch oft auf sprachlichem und Expertenwissen dieser Domäne, weshalb sie oft unangemessen bzw. mißverständlich sind und falsche Erwartungen erzeugen (vgl. Peters 1990, 153). Besonders problematisch sind sogenannte minimale Metaphern,

bei denen durch die enge Verwandtschaft von Bildspender und Bildempfänger der metaphorische Charakter verblaßt.

Das soll an einem einfachen *Beispiel* verdeutlicht werden. Die graphische Nutzeroberfläche *Windows 95* beinhaltet einen sogenannten *Aktenkoffer*. Dieser soll es ermöglichen, Dateien, an denen an verschiedenen Orten (etwa zu Hause und in der Universität) gearbeitet werden soll, leichter aktualisieren zu können. Mit dem Konzept *Aktenkoffer* verbinden sich alltagsweltlich m. E. folgende Tätigkeiten:

Einpacken von Dokumenten – Ortswechsel – Auspacken – Benutzen – Einpacken (evtl. mehr als mitgenommen wurde) – Ortswechsel zurück – Auspacken (und Einsortieren benutzter Dokumente in das Ablagesystem).

Geht man jedoch mit diesem alltagsweltlichen Verständnis an das – scheinbar intuitiv benutzbare – Hilfsmittel heran, wird man nicht den gewünschten Erfolg haben: Einmal in den „Aktenkoffer“ gepackte Dateien dürfen nämlich, wenn sie nach dem ersten Ortswechsel bearbeitet werden sollen, nicht aus dem „Aktenkoffer“ entfernt werden. Falls dies dennoch geschieht (indem man die Dateien aus dem „Aktenkoffer“ heraus auf ein anderes Speichermedium kopiert, dort bearbeitet und schließlich wieder in den Aktenkoffer zurückkopiert), funktioniert der automatische Abgleich der Daten nicht. Im Extremfall kann die (intuitive) Nutzung des Aktenkoffers zu völligem Datenverlust führen.

Eine andere Ursache von Mißverständnissen kann es außerdem sein, eingeführte Metaphern inkonsequent zu benutzen: Was nützt beispielsweise die in der *Windows 95*-Oberfläche neue und dem *Apple Macintosh* „entlehnte“ *Papierkorb-Metapher*, wenn sich die damit im Zusammenhang stehenden Tätigkeiten (im Alltagswissen: *wegwerfen/hineinwerfen* vs. *leeren*) nicht konsequent an dieser Metapher orientieren: So wird das „Wegwerfen“ einer Datei nach wie vor durch den Befehl *Löschen* ausgelöst, während die Sicherheitsabfrage „Soll Dokument1 wirklich in den Papierkorb *verschoben* werden?“ lautet.

6.2 Grammatische Ursachen für Miß- und Nichtverstehen

6.2.1 Fehlende Konnektive

Konjunktionen und Konjunkionaladverbien stellen neben der syntaktischen eine semantische Verbindung zwischen Wörtern, Wortgruppen oder Sätzen her. Fehlen sie (wie es in einer Vielzahl von Fehlermeldungen oder Dialogfenstern der Fall ist), ist der Rezipient gezwungen, die Entscheidung über die semantischen Verhältnisse durch die Rekonstruktion des Kontextes zu treffen. Dies ist jedoch – insbesondere

für Novizen – nicht immer eindeutig möglich. Noch irreführender ist es, wenn einige Fehlermeldungen *kausal* (a), andere dagegen *konsekutiv* (b) aufgebaut sind:

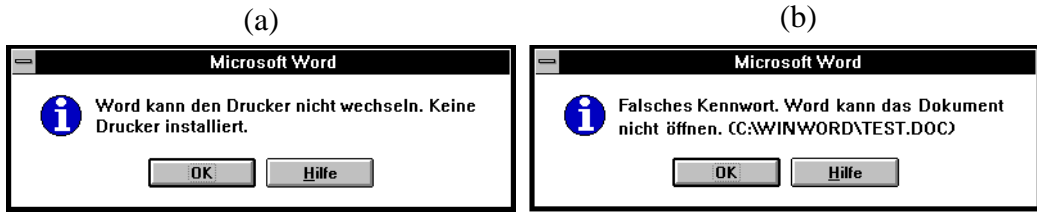


Abb. 2: Beispiele für Fehlermeldungen kausaler (a) und konsekutiver (b) Art

6.2.2 Fragen

Häufige Ursache für Unsicherheiten und Bedienungsfehler sind Dialogfenster, in denen als Antwortoptionen für Entscheidungsfragen nicht – wie vom Benutzer erwartet werden kann – *Ja* bzw. *Nein*, sondern verschiedene Kombinationen aus *OK*, *Abbrechen* und *Nein* zur Verfügung stehen. Besonders irritierend erscheint dabei folgendes Beispiel:



Abb. 3: Beispiel für irreführende Antwort-Optionen

Hier gerät der sprachliche Stimulus *Abbrechen* in Konflikt mit der Handlungsintention des Benutzers (*Abbrechen* bzw. *Beenden des Druckauftrags*): Wählt ein Benutzer, der den Druck abbrechen möchte, die ins Auge springende Antwortoption *Abbrechen*, erreicht er genau das Gegenteil: Der Druck wird fortgesetzt. Unmißverständlich wäre hier das Antwortpaar *Ja – Nein*.

6.2.3 Thema-Rhema-Struktur

Da die Mensch-Computer-Interaktion mit all ihren potentiellen „Inhalten“ und in ihrer ganzen Komplexität nicht logisch aufeinander aufbauend verläuft, tauchen zwangsläufig ständig neue sprachliche Inhalte auf – und zwar in thematischer Funktion. Das *Thema* einer Äußerung ist jedoch gewöhnlich das Bekannte, das Alte, eine präsupponierte oder kontextuell präsenste Information. Der Benutzer steht deshalb häufig vor komplexen „Äußerungen“, denen er – da sie für ihn ausschließlich rhematische Elemente enthalten – keine Bedeutung zuweisen kann.

Ein *Beispiel* von vielen, wegen dessen ich im Laufe der Zeit mehrfach von Kolleginnen und Kollegen um Hilfe gebeten wurde:

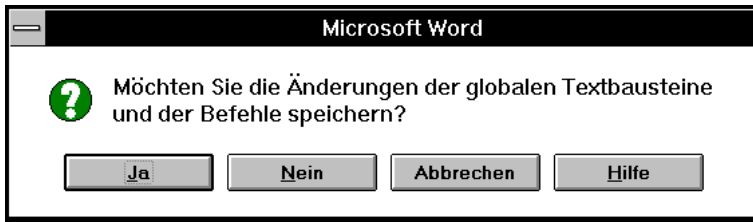


Abb. 4: Beispiel für problematische Thema-Rhema-Struktur

Nicht nur ein Anfänger weiß oft nicht, was es mit diesen *globalen Textbausteinen* (dazu zählen z. B. Menü- und Symbolleisteneinstellungen und Makros) auf sich und schon gar nicht, daß er diese verändert hat. Er/sie steht jetzt vor der Wahl, die System-Anfrage (mit nicht bekannten Implikationen) „intuitiv“ zu beantworten oder sich – was meist zeitaufwendig ist und Energie kostet – zusätzliche Informationen (Erklärungen, Hilfen) zu verschaffen, um adäquat handeln zu können.

Dieses Problem der Thema-Rhema-Gliederung ist in ähnlicher Form auch aus Hypertexten bekannt: Einzelne Knoten müssen relativ autonom verständlich sein oder dem Leser zumindest Möglichkeiten bieten, sich Informationen, auf die aufgebaut wird, zu beschaffen (vgl. Todesco 1997). Beides ist jedoch im *Beispiel* nicht bzw. schlecht realisiert: Wenn der Nutzer lediglich einen Button zur Symbolleiste hinzugefügt hat, könnte die Systemanfrage dementsprechend konkret verfaßt sein (was natürlich einen erhöhten programmiertechnischen Aufwand verlangt). Andererseits wäre es möglich und wesentlich benutzerfreundlicher, erklärende Definitionen in sogenannten *pop-up-Fenstern* (also Fenstern, die direkt über dem klärungsbedürftigen Terminus erscheinen) anzubieten, so daß der Nutzer nicht erst zwangsweise in eine andere (Hyper-)Textwelt (der Online-Hilfe) entführt wird.

6.3 Pragmatische Ursachen

Bevor an dieser Stelle, wie in den vorangegangenen Abschnitten, konkrete *Beispiele* aufgeführt werden, sollen einige prinzipielle Ursachen für das Auftreten mißlingender Interaktionsprozesse zumindest benannt werden.

Turn-taking und feedback

Die Mensch-Computer-Interaktion basiert – wie oben bereits festgestellt – in weiten Teilen auf der Metapher des *Dialogs*. Dabei treten jedoch in dieser Interaktion Differenzen in den Regularitäten und Mechanismen zur zugrundeliegenden Mensch-Mensch-Kommunikation auf, die oft gar nicht mehr wahrgenommen werden. So ist beispielsweise der Prozeß des *turn-taking* (also des Sprecherwechsels zwischen miteinander Kommunizierenden) oft dadurch gestört, daß entsprechende Signale für den Benutzer, den *turn* zu übernehmen, fehlen, nicht wahrgenommen bzw. mißverstanden werden.

Damit im Zusammenhang stehen auch Probleme des *feedbacks* durch den Computer: Die meisten sprachlichen Handlungen des Benutzers zeitigen ein nicht-sprachliches *feedback*, welches in den extremsten Fällen noch nicht einmal wahrgenommen werden kann, wie etwa beim *Kopieren* oder *Ausschneiden* von Text: Das Gelingen der Aktion kann erst beim *Einfügen* überprüft werden; als kognitive Belastung kommt hinzu, daß sich der Benutzer über den gesamten Vorgang hinweg (z. B. beim Verschieben von Textteilen über viele Seiten) den Inhalt des in der Zwischenablage befindlichen Textes merken muß. Ein direkter Zugriff auf die Zwischenablage, welche ja die zuvor kopierten bzw. ausgeschnittenen Objekte enthält, ist unter *Windows* aus Textverarbeitungsprogrammen heraus nicht möglich. (Eine kleine Anmerkung: Auch beim *Ausschneiden* und *Einfügen* von Text macht sich ungenaue Metaphorik bemerkbar: Unerfahrene Nutzer zeigen sich oft davon überrascht, daß *ausgeschnittener* Text nach dem *Einfügen* – im Gegensatz zur bekannten Arbeit mit Schere, Papier und Klebstoff – weiterhin zur Verfügung steht, also mehrfach eingefügt werden kann.)

Mündlichkeit – Schriftlichkeit

Menschliche dialogische Kommunikation ist – wird von neueren Formen wie dem *Online-Chat* (über Tastatur und Bildschirm stattfindende „Gespräche“ im Internet zwischen zwei oder mehreren Partnern) abgesehen – im Bereich der Mündlichkeit angesiedelt. Computer-„Dialoge“ laufen jedoch über geschriebene (bzw. visuell präsentierte) Sprache ab. Die Tatsache, daß „mündliche und schriftliche Formulierungen beträchtlich voneinander abweichen“ können, „wird bei der Mensch-Maschine-Kommunikation bisweilen übersehen“ (Fellbaum 1992, 157). Der „höhere strukturelle Komplexitätsgrad der Schriftsprache“ (Leont’ev 1974, 24), der Wegfall verständnissichernder nonverbaler Mittel und das Fehlen der die mündliche Sprache regulierenden Dynamik (vgl. Lurija 1982, 241) sowie insbesondere die Mischung aus Elementen von *Mündlichkeit* (insbesondere in Dialogfenstern, Menüleisten und -befehlen) und *Schriftlichkeit* (insbesondere in Online-Hilfen) führen zu einem prinzipiell erhöhten Potential an Miß- bzw. Nichtverstehen.

6.3.1 Indirekte Sprechakte

Eine Hauptursache für Miß- bzw. Nichtverstehen in der Mensch-Computer-Interaktion stellen *indirekte Sprechakte* dar. Computer-Benutzer sind – auf Grund mangelnden Hintergrundwissens – oft nicht in der Lage, indirekte Sprechakte bzw. deren Illokution zu erkennen. So ist beispielsweise die Fehlermeldung in Abb. 5 von ihrer sprachlichen Form her rein *deklarativ*. Ein unerfahrener Nutzer versteht lediglich, daß er etwas falsch gemacht hat. Das und gegebenenfalls was er dagegen tun muß, um die Ursache dieser Fehlermeldung zu beseitigen, wird nicht explizit verbalisiert. Für den Systemdesigner bzw. für einen erfahrenen Nutzer hingegen ist dieselbe Fehlermeldung eine unmißverständliche *Aufforderung*, entweder den

Namen des zu sichernden Dokuments oder den Platz in der Verzeichnisstruktur zu ändern.



Abb. 5: Beispiel für Fehlermeldung mit indirektem Sprechakt

Das (kurze wie verständliche) „Hilfe“ mit der dahinterstehenden Online-Hilfe ist nur *ein* Ausweg, und dazu einer, der von den Benutzern ungern beschritten wird: Von den fünf Versuchgruppen nutzte nur eine (und zwar die mit der meisten Computer-Erfahrung) die Online-Hilfe des Systems, obgleich in der Aufgabenstellung explizit dazu aufgefordert wurde, „gegebenenfalls die Hilfen, die das System bietet“, zu verwenden! Eine Probandin, nach dem Versuch daraufhin befragt, antwortete lapidar, daß sie diese ohnehin nicht verstanden hätte...

6.3.2 Verletzung Gricescher Konversationsmaximen

Wird die Interaktion zwischen Mensch und Computer auf der Dialog-Metapher gegründet, scheint es auch berechtigt zu sein, Grice' Konversationsmaximen (1975; dt. 1979) für diese zu reklamieren (vgl. Brennan 1990). Je häufiger nämlich Erwartungshaltungen der Benutzer in bezug auf Interaktionssequenzen nicht entsprochen wird, desto weiter entfernt sich die Interaktion von der angestrebten Dialoghaftigkeit. Die sprachliche Gestaltung moderner Interfaces verletzt jedoch mindestens folgende Maximen und provoziert dadurch mißlingende Interaktionsprozesse.

Verletzung der Maximen der Quantität

Je mehr Informationen das System liefert, um so größer ist die kognitive Belastung des Benutzers und um so wahrscheinlicher wird es demzufolge, daß durch eine notwendigerweise höhere Selektivität Informationen ausgeblendet werden, die für eine reibungslose Interaktion notwendig sind. Insofern stellen viele Dialogfenster wie in Abb. 6, eine Verletzung der Maxime „Mache Deinen Beitrag nicht informativer als nötig.“ (Grice 1979, 249) dar.

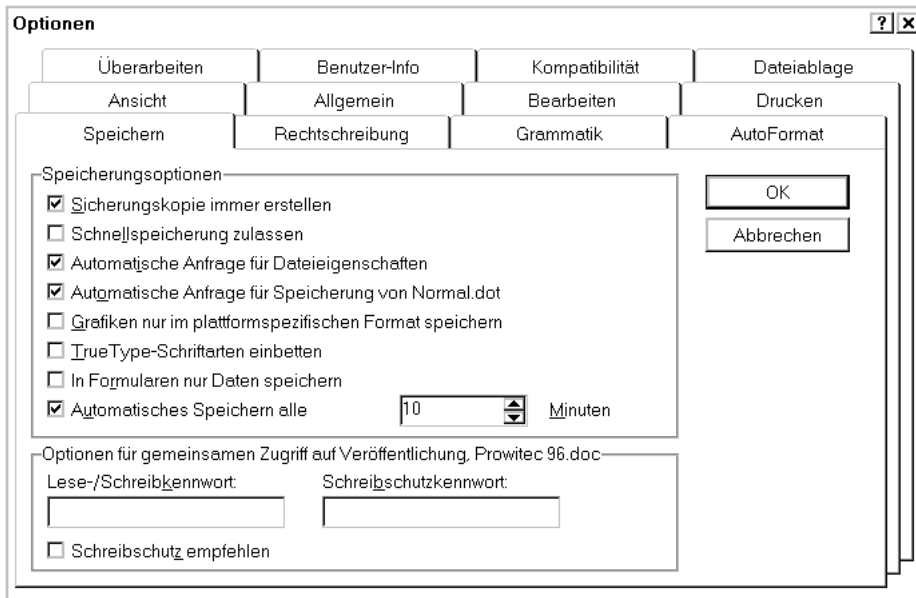


Abb. 6: Beispiel für ein überfülltes Dialogfenster

Häufiger jedoch werden dem Nutzer zu wenige Informationen zur Verfügung gestellt und damit gegen die Maxime „Mache Deinen Beitrag so informativ (wie für die gegebenen Gesprächszwecke) nötig.“ (ebd.) verstoßen.



Abb. 7: Beispiel für mangelnde Informativität

In beiden Fällen wird der Benutzer gezwungen, sich intensiv mit dem Interface auseinanderzusetzen, was von der eigentlich zu bewältigenden Aufgabe wegführt. Überprüfen Sie zum Beispiel selbst, ob Sie auf Anhieb sagen könnten, auf welchen „Karteikarten“ des Dialogfensters *Optionen* (s. oben) welche Informationen zu finden bzw. welche Optionen der sichtbaren „Karte“ *Speichern* Ihnen verständlich sind. Man beachte, daß von den „Karteikarten“ keine zusätzlichen Hilfen angeboten werden bzw. von dort aus erreichbar sind.

Verletzung der Maxime der Relevanz

Folgende Fehlermeldung wurde angezeigt, als eine Probandengruppe die in den Text einzufügende Grafik über das Grafik-Modul *Microsoft Draw* bearbeiten wollte:



Abb. 8: Beispiel für irrelevante Meldung

Diese Fehlermeldung ist nur für Experten verständlich, für den durchschnittlichen Nutzer hingegen in dieser Form sogar irrelevant: Es wird etwas vom System ignoriert, von dessen Existenz der durchschnittliche Anwender ohnehin nichts weiß, eine Entscheidungsmöglichkeit seitens der Nutzer besteht nicht. Den Probandinnen blieb also lediglich die Möglichkeit, die Meldung zu bestätigen, woraufhin sie ihre Arbeit – ohne Beeinträchtigungen (!) – fortsetzen konnten.

Verletzung der Maximen der Modalität

Fehlermeldungen folgender Art sind nicht die seltensten und “seem like either dire digital disasters inside the system or messages designed for somebody else” (Brennan 1990, 397):

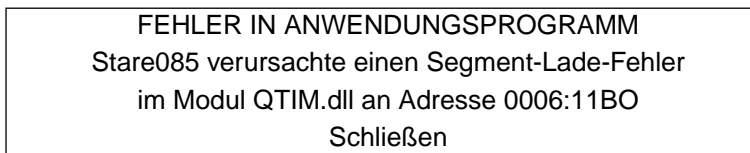


Abb. 9: Beispiel für Unverständlichkeit

Daß derartige Fehlermeldungen die Maxime „Vermeide Dunkelheit des Ausdrucks.“ (Grice 1979, 249) verletzen, ist schon trivial.

7 Auswege

Eine Vielzahl der Interaktionsprobleme erwächst aus der überfrachteten Funktionalität heutiger Software-Pakete. Van Berkel und Donga (1994) haben beispielsweise für das Textverarbeitungssystem *WordPerfect 5.1*TM gezeigt, daß selbst erfahrene Computer-Nutzer nur einen Bruchteil der zur Verfügung stehenden Funktionen verwenden. Werkzeuge – insbesondere technische – werden im Laufe ihrer Entwicklung gewöhnlich immer besser auf spezielle Aufgaben eingestellt. Im Bereich der Software-Entwicklung hingegen wird versucht, Universal-Werkzeuge zu schaffen, mit denen alle erdenklichen Schreib-Aufgaben erledigt werden können. Das ist in etwa so, als ob es für filigrane Goldschmiedearbeiten, das Einschlagen eines Nagels oder Gleisbauarbeiten nur eine Sorte Hammer gäbe.

Es erscheint deswegen sinnvoll, nutzer- und aufgabenspezifische Software-Umgebungen zu entwickeln. Diese könnten (und in Anbetracht ökonomischer Tatsachen

sollten) durchaus auf vorhandenen Applikationen aufsetzen, könnten jedoch – je nach konkretem Einsatzgebiet – variabel sein in bezug auf

- die Gesamtanzahl der verfügbaren Funktionen,
- die Auswahl der notwendigen Funktionen,
- die (fachspezifische) Bezeichnung der Funktionen,
- die (individuelle) Schwerpunktsetzung auf sprachliche oder graphische Interface-Elemente,
- die Art und den Umfang der Hilfen sowie
- die Art des (anstelle des Standard-Thesaurus') zu implementierenden Fachwörterbuchs.

Besonders hervorzuheben, sind Bemühungen, den Textproduktionsprozeß durch spezielle interaktive Autorenumgebungen, wie z. B. *Composer* (vgl. Shurville/Hartley/Pemberton 1997), zu unterstützen. Neben den notwendigen Basis-Elementen des *Interfaces* könnten hier jedoch die metasprachlichen Elemente, die den Schreiber z. B. bei der Textorganisation unterstützen sollen, zum zusätzlichen Problem werden. Der Erfolg, der dem Projekt gewünscht wird, bleibt abzuwarten.

Ein Schritt in die richtige Richtung sind auch sogenannte *Adaptive User Interfaces*, die sich auf der Basis einer Analyse des Nutzerverhaltens an den Benutzer anpassen sollen (vgl. z. B. Schneider-Hufschmidt/Kuehme/Malinowski 1993).

Alle diese Möglichkeiten scheinen jedoch nur bis zu einem gewissen Punkt erfolgversprechend. Insbesondere durch die sich über Jahre und inzwischen schon Jahrzehnte hinstreckende Entwicklung einiger Applikationen, bei denen neuere Versionen auf jeweils vorhergehenden basieren, scheinen sich spezifische Diskurs- und Sprachverwendungsmuster herauszubilden, die für Neulinge in dieser Domäne un- bzw. mißverständlich sind. Nur durch verstärkten Einsatz aller Methoden des *usability engineering* während des gesamten Entwicklungszyklus' einer Software kann dieses Phänomen zugunsten einer erhöhten Benutzerfreundlichkeit zurückgedrängt werden (vgl. Lutz 1996). Dazu ist es auch notwendig, das Instrumentarium des *usability engineering*s zu verfeinern und – was bisher zu wenig geschehen ist – insbesondere durch Methoden der Angewandten Sprachwissenschaft zu ergänzen.

Literatur

- Antos, Gerd (1989): Textproduktion. Ein einleitender Überblick. In: Antos, Gerd/ Krings, Hans P. (Hrsg.): Textproduktion. Ein interdisziplinärer Forschungsüberblick. Tübingen: Niemeyer [Konzepte der Sprach- und Literaturwissenschaft; 48], 5-57
- Bangert-Drowns, Robert L. (1993): The Word Processor as an Instructional Tool. A Meta-Analysis of Word Processing in Writing Instruction. In: Review of Educational Research 1 (63), 69-93
- de Beaugrande, Robert (1984): Text Production. Toward a Science of Composition. Norwood NJ: Ablex [Advances in Discourse Processes; IX]
- van Berkel, Arrie/ Donga, Margaret (1994): Problem Solving Strategies used by Experienced WP5.1-Users. In: van Waes, Luuk/ Woudstra, Egbert/ van den Hoven, Paul (eds.): Functional Communication Quality. Amsterdam, Atlanta: Rodopi [Utrecht Studies in Language and Communication; 4], 195-207

- Blum-Kulka, Shoshana/ Weizman, Elda (1988): The Inevitability of Misunderstandings. *Discourse Ambiguities*. In: *Text. An Interdisciplinary Journal for the Study of Discourse* 3 (8), 219-241
- Brennan, Susan E. (1990): Conversation as Direct Manipulation. In: Laurel, Brenda (eds.): *The Art of Human-Computer Interface Design*. Reading MA: Addison-Wesley, 393-404
- Eco, Umberto (1988): *Das Foucaultsche Pendel*. München: Hanser
- Erickson, Thomas D. (1990): Working with Interface Metaphors. In: Laurel, Brenda (eds.): *The Art of Human-Computer Interface Design*. Reading MA: Addison-Wesley, 65-73
- Fellbaum, Klaus (1992): Anwendungsaspekte der sprachlichen Mensch-Maschine-Kommunikation. In: Mangold, Helmut (Hrsg.): *Sprachliche Mensch-Maschine-Kommunikation*. München: Oldenbourg Verlag, 153-166
- Floyd, Christiane (1990): Leitbilder für die Gestaltung interaktiver Systeme. Computer sind keine Menschen. In: Endres-Niggemeyer, Brigitte (Hrsg.): *Interaktion and Kommunikation mit dem Computer. Jahrestagung der Gesellschaft für linguistische Datenverarbeitung*. Berlin u. a.: Springer, 12-21
- Grice, Herbert Paul (1979): Logik und Konversation. In: Meggle, Georg (Hrsg.): *Handlung, Kommunikation, Bedeutung*. Frankfurt/Main: Suhrkamp, 243-265
- Jakobs, Eva-Maria (1995): Text und Quelle. Wissenschaftliche Textproduktion unter Nutzung externer Wissensspeicher. In: Jakobs, Eva-Maria/ Knorr, Dagmar/ Molitor-Lübbert, Sylvie (Hrsg.): *Wissenschaftliche Textproduktion. Mit und ohne Computer*. Frankfurt/Main u. a.: Lang, 91-112
- Jakobs, Eva-Maria/ Knorr, Dagmar (1995): Wissenschaftliches Schreiben am Computer. Ein professionelles Muß? In: *OBST (Osnabrücker Beiträge zur Sprachtheorie)* 50, 83-106
- Knorr, Dagmar (1995): Elektronische Medien im wissenschaftlichen Alltag. Auswirkungen auf die persönliche Literaturverwaltung und -nutzung. In: Jakobs, Eva-Maria/ Knorr, Dagmar/ Molitor-Lübbert, Sylvie (Hrsg.): *Wissenschaftliche Textproduktion. Mit und ohne Computer*. Frankfurt/Main u. a.: Lang, 53-71
- Krings, Hans P. (1992): Schwarze Spuren auf weißem Grund. Fragen, Methoden und Ergebnisse der Schreibprozeßforschung im Überblick. In: Krings, Hans P./ Antos, Gerd (Hrsg.): *Textproduktion. Neue Wege der Forschung*. Trier: Wissenschaftlicher Verlag Trier [Fokus. Linguistisch-Philologische Studien; 7], 45-110
- Leont'ev, Aleksej A. (1974): *Psycholinguistik und Sprachunterricht*. Stuttgart, Berlin, Köln, Mainz: Kohlhammer
- Lurija, Aleksandr R. (1982): *Sprache und Bewußtsein. Beiträge zur Psychologie*. Berlin: Volk und Wissen
- Lutz, Benedikt (1996): Zur Anwenderfreundlichkeit computerbasierter Technischer Dokumentation. Die Textlinguistik als missing link zwischen Hypertext-Forschung und Usability engineering? In: Ensink, Titus/ Sauer, Christoph (eds.): *Researching Technical Documents*. Groningen: Department of Speech and Communication [Groningen Series on Language Use and Communication GLUC; 2], 87-110
- Molitor-Lübbert, Sylvie (1989): Schreiben und Kognition. In: Antos, Gerd/ Krings, Hans P. (Hrsg.): *Textproduktion. Ein interdisziplinärer Forschungsüberblick*. Tübingen: Niemeyer [Konzepte der Sprach- und Literaturwissenschaft; 48], 278-296
- Norman, Kent L. (1991): *The Psychology of Menu Selection. Designing Cognitive Control at the Human/Computer Interface*. Norwood NJ: Ablex
- Peters, Wolfram (1990): *Didaktik der Informationsverarbeitung. Eine sprachdidaktisch-semiotische Analyse der Mensch-Maschine-Kommunikation*. Tübingen: Niemeyer
- Plieninger, Martin (1991): *Computereinsatz im Aufsatzunterricht. Rechnergestützte Analyse von Gebrauchstexten und Möglichkeiten des Schreibens in der Sekundarstufe I unter FRAMEWORK III*. Stuttgart: Metzler
- Rahmenrichtlinien Deutsch für Sekundarschulen in Sachsen-Anhalt, hrsg. vom Kultusministerium des Landes Sachsen-Anhalt. Magdeburg: Garloff 1995
- Rosch, Eleanor/ Mervis, Carolyn B. (1975): Family Resemblances. Studies in the Internal Structure of Categories. In: *Cognitive Psychology* 7, 573-605

-
- Sagawe, Helmuth (1989): Kommunikationspartner „Computer“. In: *Wirkendes Wort* 2 (39), 294-305
- Schmitz, Ulrich (1995): Geistiges Sammelsurium in technischer Perfektion. Schreibbarock und Schreibaskese am Computer. In: Jakobs, Eva-Maria/ Knorr, Dagmar/ Molitor-Lübber, Sylvie (Hrsg.): *Wissenschaftliche Textproduktion. Mit und ohne Computer*. Frankfurt/Main u. a.: Lang, 149-168
- Schneider-Hufschmidt, Matthias/ Kuehme, Thomas/ Malinowski, Uwe (1993): *Adaptive User Interfaces. Principles and Practise*. Amsterdam et al.: North-Holland
- Shurville, Simon/ Hartley, Anthony/ Pemberton, Lyn (1997): A Development Methodology for Composer. Computer Support Tool for Academic Writing in a Second Language. In diesem Band, 171-182
- Siefkes, Dirk (1989): Beziehungskiste Mensch-Maschine. In: *Sprache im technischen Zeitalter* (27), 332-343
- Todesco, Rolf (1997): Die Definition als Textstruktur im Hyper-Sachbuch. In diesem Band, 109-120
- van Waes, Luuk (1994): Computers and Writing. In: Pogner, Karl-Heinz (eds.): *More About Writing*. Odense: Odense University [Odense Working Papers in Language and Communication; 6], 41-61
- Wagner, Jörg (1995): Informations- und kommunikationstechnologische Grundbildung. Überlegungen zum Beitrag des Faches Deutsch und Vorschläge zur Arbeit in den Lernbereichen. In: Landesinstitut für Lehrerfortbildung, Lehrerweiterbildung und Unterrichtsforschung in Sachsen-Anhalt (Hrsg.): *Aufsätze über einen Transfermodellversuch „IKG an Gymnasien“*. [Schwerpunktfach Deutsch], 9-50
- Weingarten, Rüdiger (1989): Die Verkabelung der Sprache. Grenzen der Technisierung von Kommunikation. Frankfurt/Main: Fischer