

Ein empirischer Vergleich einer desktop- mit einer menü-orientierten Benutzungsoberfläche für ein relationales DBMS*.

Matthias Rauterberg
Angewandte Informatik
Universität Oldenburg

Zusammenfassung

Da die Vor- und Nachteile einer direkt-manipulativen Benutzungsoberfläche noch sehr kontrovers diskutiert werden, wurde in dieser Studie ein empirischer Vergleich zwischen einer direkt-manipulativen und einer konventionellen, menü-orientierten Benutzungsoberfläche eines DBMS durchgeführt. 12 Benutzer (6 Experten mit der direkt-manipulativen DBMS-Oberfläche; 6 Experten mit der menü-orientierten DBMS-Oberfläche) mußten zehn Standardaufgaben bearbeiten. Die Vorerfahrung der Experten im Umgang mit EDV insgesamt betrug 3.700 Std. (direkte Manipulation), bzw. 7.500 Std. (menü-orientierte Oberfläche) und wurde über einen 115-Item Fragebogen sorgfältig erhoben. Als Ergebnis zeigte sich eine statistisch bedeutsame Überlegenheit der direkt-manipulativen Benutzungsoberfläche mit der "Maus" über die konventionelle, menü-orientierte Oberfläche mit der "Funktionstaste". Die Experten mit der direkt-manipulativen DBMS-Oberfläche benötigten trotz ihrer geringeren Vorerfahrung im Durchschnitt über alle Aufgaben hinweg nur 55% der Bearbeitungszeit gegenüber den Experten der konventionellen, menü-orientierten DBMS-Oberfläche. Es zeigte sich darüber hinaus ein bedeutsamer Zusammenhang zwischen der Art der Aufgabe und dem Typ der DBMS-Oberfläche.

1. Einleitung

In dieser Studie geht es darum, herauszufinden, welcher Typ von Benutzungsoberfläche - "desktop" oder "menü-orientiert" - für die Benutzung eines relationalen Datenbank-Management-System (DBMS) am geeignetesten erscheint. In einer Detailanalyse sollen die besonderen Eigenschaften der überlegeneren Benutzungsoberfläche vorgestellt werden.

2. Stand der Forschung

Die Vorteile von Desktop-Oberflächen scheinen so offensichtlich zu sein, daß es kaum experimentelle Untersuchungen zu finden gibt, die die Überlegenheit der Desktop-Oberflächen gegenüber den anderen Arten des Dialoges aufzeigen (SHNEIDERMAN 1987; SMITH & MOSIER, 1986; KRAUSE, 1986). So fordern dann auch HUTCHINS, HOLLAN & NORMAN (1986, S. 123) eine eingehendere empirische Evaluation der direkt-manipulativen Oberfläche.

Bei der Gestaltung der Benutzungsoberfläche wird für die ergonomisch adäquate Repräsentation des Dialog- und Anwendungszustandes Platz auf der E/A-Schnittstelle benötigt. Um diesen physikalisch begrenzten Platz optimal auszunutzen, wurden verschiedene Dialogarten entwickelt.

* Diese Studie ist im Rahmen des vom BMFT geförderten Forschungsprojektes "Benutzerorientierte Software-Entwicklung und Dialoggestaltung" am Lehrstuhl für Arbeits- und Organisationspsychologie der ETH-Zürich im Verbund mit der ADI-GmbH aus Karlsruhe durchgeführt worden.

ALTMANN (1987) verglich eine direkt-manipulative (McWrite) und eine kommando-orientierte (WordStar) Benutzungsoberfläche hinsichtlich ihrer Erlernbarkeit und Handhabbarkeit für Novizen. Novizen mit McWrite schnitten beim Erlernen deutlich besser ab als Novizen mit WordStar (siehe ebenso ROBERTS & MORAN; 1983). Zu einem scheinbar entgegengesetzten Ergebnis kommen WHITESIDE et.al. (1985); dieses Ergebnis kann dadurch zustande gekommen sein, daß die besonders vorteilhafte, kommando-orientierte Version mit einem Online-Ausgabefenster aller aktuell gültigen Befehle nicht als menü-orientiert eingestuft worden ist.

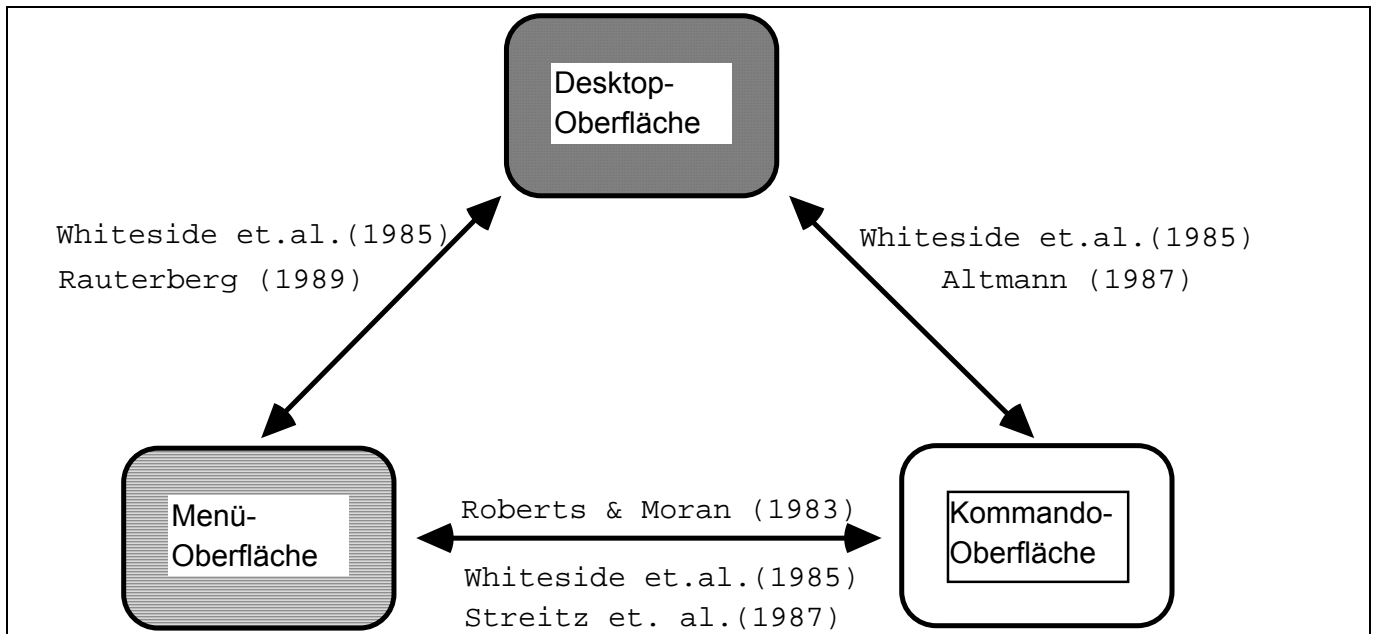


Abbildung 1: Übersicht über die unterschiedlichen empirischen Studien zum Vergleich der drei am häufigsten vorkommenden Arten von Benutzungsoberflächen: kommando-, bzw. menü-orientiert und direkt-manipulativ.

ROBERTS & MORAN (1983) stellten bei Novizen und Experten die Überlegenheit einer Reihe von Texteditoren mit menü-orientierten Oberflächen über diejenigen mit einer kommando-orientierten Oberfläche fest. HAUPTMANN & GREEN (1983) fanden keinen signifikanten Unterschied bei der Bearbeitungszeit für die Erstellung von Grafiken zwischen einer menü- und einer kommando-orientierten Oberfläche bei Novizen und Experten. Allerdings zeigte sich ein Vorteil für die menü-orientierte Oberfläche bei der Anzahl der benötigten Dialogoperatoren. Dem entgegen war bei STREITZ et.al. (1987) (instruierte Novizen) und WHITESIDE et.al. (1985) (Novizen, gelegentliche Benutzer und Experten) die Kommando-Oberfläche besser.

Bei WHITESIDE et.al. (1985) ist die menü-orientierte Oberfläche der Desktop-Oberfläche überlegen. Beachtenswerterweise sprechen WHITESIDE et.al. (1985) jedoch von ikonien-orientierten statt von direkt-manipulativen Oberflächen, so daß nur schwer abzuschätzen ist, in wie weit dieser Unterschied für das Ergebnis entscheidend ist.

Zurecht betonen WHITESIDE et.al. (1985), daß die aufgabenangemessene Gestaltung der Benutzungsoberfläche sehr wichtig, zum Teil sogar wichtiger als die Art der Benutzungsoberfläche ist.

3. Fragestellungen

Da bisher nur unzureichende und widersprüchliche Ergebnisse zum Vergleich von direkt-manipulativen mit konventionellen, menü-orientierten Benutzungsoberflächen vorliegen, sollen diese beiden Dialogarten im Rahmen dieser Studie verglichen werden:

menü-orientierten Benutzungsoberfläche (ASCII): die konventionelle, ascii-orientierte Benutzungsoberfläche, bei der die Dialogführung **mit Funktionstasten** (inklusive Cursor-Steuertasten) und Auswahlmenüs abgewickelt wird;

desktop-orientierte Benutzungsoberfläche (MAUS): die grafik-orientierte, direkt-manipulative Benutzungsoberfläche, bei der die Dialogführung **mit der Maus** durch Anklicken von maus-sensitiven Bereichen vollzogen wird.

Es wird in dieser Arbeit also nicht ein Vergleich zwischen einer direkt-manipulativen und einer kommando-orientierten Oberfläche (wie z.B. dBASE, SQL, etc.) durchgeführt (siehe Abbildung 1).

Im folgenden wird zunächst die ASCII-Oberfläche und dann die MAUS-Oberfläche genauer vorgestellt. Die ASCII-Oberfläche besteht aus verschiedenen Modulen (z.B. Daten, Liste, Wahl, Rechnen, etc.), zwischen denen der Benutzer über das strikt hierarchische Hauptmenü durch Eingabe von Buchstaben in der "Befehls"-Zeile hin- und herwechseln kann. Innerhalb der einzelnen Module erfolgt die Auswahl der jeweils gewünschten Routine (z.B. im Modul DATEN: Eingabe, Korrektur, Loeschen, Ausgabe, Halt >[]) ebenfalls über die Eingabe der entsprechenden Anfangsbuchstaben (E,K,L,A) (siehe Abbildung 2). Während der Abarbeitung einer Routine stehen dem Benutzer jedoch nur noch die Funktionstasten (inklusive Cursor-Steuerung) für die weitere Dialogsteuerung zur Verfügung (siehe Abbildung 3).

```

CAMP


---


ADI --- ADIMENS.EXEC V2.21e 29.07.88

ADI Software GmbH --- Datenbanksystem : ADIMENS
(c) Copyright by ADI Software GmbH 1983-88

Datenbasis = ..\DB-CAMP.211\CAMP ...

Adressen      Platz      Gruppe



---


Befehl: Daten Liste Wahl Rechnen Verbund Tedi Gruppe Info Halt >[ ]
1HILFE 2SCHALT 3DATEI 4          5          6          7EINGAB 8AUSGAB 9 10

```

Abbildung 2: Übersicht über die Start-Maske nach Laden der Datenbank CAMP mit den drei Dateien "Adressen", "Platz" und "Gruppe". Die vorletzte Bildschirmzeile ist die Hauptmenü-Leiste; die Benutzer-Eingabe erfolgt hinter ">"; die letzte Bildschirmzeile gibt die aktuelle Belegung der Funktionstasten an.

Um nun aus dem Hauptmenü in die Routine "Ausgabe" des Moduls "Daten" zu gelangen, muß der Benutzer die beiden Buchstaben "D" und "A" hintereinander eingeben. Nachdem eine leere Maske entsprechend der aktuellen Datei erschienen ist, kann nun der Benutzer den gewünschten Suchbegriff (einschließlich der "Wildcards") in das leere Merkmalsausprägungsfeld eines der vorgegebenen Schlüsselmerkmale eintragen. Ist mindestens ein Datensatz in der Datei vorhanden, auf den dieser Suchbegriff paßt, so erfolgt die Ausgabe des ersten gefundenen Datensatzes als Maske auf dem Bildschirm. Dieser Zugriff auf eine Menge von Datensätzen wird "Selektion" genannt. Zusätzlich besteht aber auch bei beiden Oberflächen die Möglichkeit, für jedes vorhandene - also auch Nicht-Schlüssel - Merkmal eine logische Bedingung zu definieren. Diese einzelnen Bedingungen lassen sich durch logische Operatoren (UND, ODER, >, ≥, <, ≤, =, ≠) zu komplexeren Filtern zusammensetzen; diese Filter werden in dem Modul "Wahl" erstellt und gelten für die jeweils aktive Datei. Der Unterschied zur Selektion besteht darin, daß die Selektion jeweils nur für ein vorgegebenes Schlüsselmerkmal definiert werden kann und auch nur für die folgenden Arbeitsschritte in der jeweiligen Routine erhalten bleibt. Diese beiden Funktionalitäten - Selektion und Wahl - sind bei beiden Oberflächen dem Benutzer zugänglich.

<u>CAMP.Adressen (CON,CON) (- - -)</u>	<u>Daten.Ausgabe</u>
Namensschlüssel:>D..Rauterberg, Matthias_____	
Anrede: Herrn_	
Vorname: Matthias_____ Nachname: Rauterberg_____	
Straße: Kurvenstraße 1_____	
Ort: 8006 Zürich_____ Land: Schweiz_____	
Tel: -_____	
Res.-Wunsch erh.: __/__/__ Anreise: __/__/__ Abreise: __/__/__	
Platz-Angebot: _____ Platz-Angebot abges.: __/__/__	
Res.-Bestät. erh.: __/__/__ Res.-Betrag erh.: __/__/__	
Bemerkungen: war noch nie auf diesem Campingplatz_____	

1FELD 2RICHTG 3DATEI 4MERKE 5 6 7RECHNE 8DRUCK 9FERTIG 10STOP	

Abbildung 3: Übersicht über die Maske des Datensatzes mit dem Primärschlüssel "Namensschlüssel" aus der Datei "Adressen" in dem Modul DATEN und in der Routine AUSGABE.

Zur Beendigung des jeweiligen Aufenthaltes in einer Routine muß der Benutzer eine der beiden Funktionstasten: "9FERTIG" oder "10STOP" drücken; hierbei steht "9FERTIG" für die normale Beendigung und "10STOP" für das vorzeitige Abbrechen. Anschließend befindet sich der Benutzer auf der Modul-Ebene. Durch Eingabe von "H" für "Halt" gelangt man zur Hauptmenü-Ebene.

Sehen wir uns nun dagegen die MAUS-Oberfläche genauer an, so können die einzelnen Module und die in ihnen enthaltenen Routinen direkt über ein Pulldown-Menü durch ein entsprechendes Verschieben der Maus erreicht werden (siehe Abbildung 4).

Ebenso wie bei der ASCII-Oberfläche erscheint nach dem Anklicken der Menü-Option "Anzeigen" in dem Pulldown-Menü "Daten", sofern zuvor in dem Pulldown-Menü "Schalter" der entsprechende

Schalter auf "Ausgabe als Maske" gesetzt wurde, zunächst eine Maske (hier "Dialogbox" genannt), in welche der benötigte Suchbegriff (einschließlich der "Wildcards") in das leere Merkmalsausprägungsfeld des jeweils gewünschten, vorgegebenen Schlüsselmerkmals eingetragen werden kann. Anschließend erfolgt die Ausgabe der gefundenen Datensätze sehr ähnlich wie bei der ASCII-Oberfläche (siehe Abbildung 5).

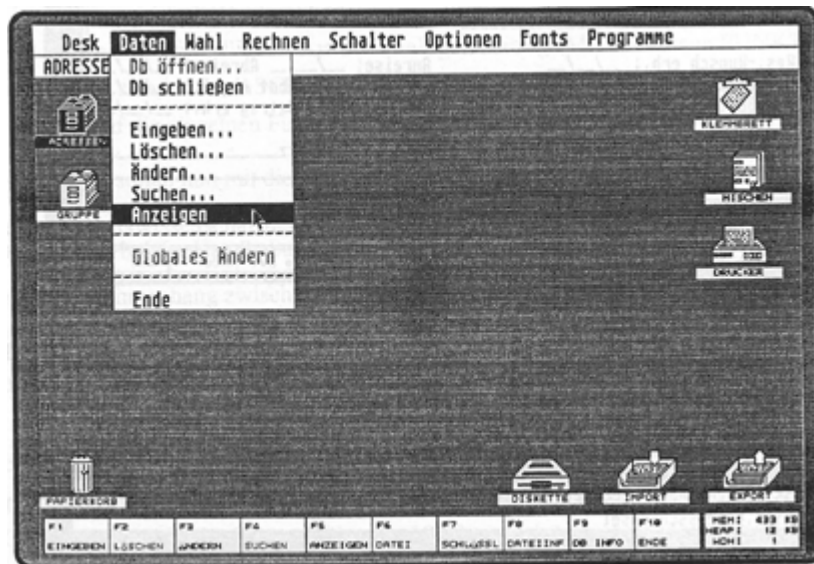


Abbildung 4: Übersicht über den "Desktop" der MAUS-Oberfläche nach dem Aufklappen des Pull-down-Menüs "Daten" und dem Anklicken der Menü-Option "Anzeigen". Im unteren Bildschirmbereich wird die aktuelle Belegung der Funktionstasten angezeigt; diese Felder sind auch zusätzlich selbst wiederum maus-sensitive Bereiche.

In einer Dialogbox erfolgt die weitere Dialogsteuerung ausschließlich durch Anklicken eines der vorhandenen Dialogknöpfe (siehe Abbildung 5: "VOR", "ABBRUCH" und "ZURÜCK"). Im Gegensatz zur Dialogbox erlaubt die Ausgabe von Datensätzen in einem Fenster (siehe Abbildung 6) die parallele Benutzung der zusätzlich auf der Desktop-Oberfläche noch direkt erreichbaren Funktionen. Diese besondere Eigenschaft der Fenster erlaubt es dann dem Benutzer, sehr flexibel seine einzelnen Arbeitsschritte zu planen und durchzuführen. Bei der ASCII-Oberfläche konnte dagegen in dem Modul "Liste" nur eine Liste definiert, erstellt und gegebenenfalls ausgedruckt werden; ein direkter Zugriff auf einzelne Datensätze aus der erstellten Liste heraus war nicht möglich.

Bei der "Ausgabe als Liste" der Datensätze einer Datei werden diese in einem Fenster nach dem voreingestellten Schlüsselmerkmal sortiert ausgegeben. Jeder Datensatz erscheint in einer Zeile, wobei die einzelnen Merkmalsausprägungen hintereinander in der jeweiligen Zeile ausgegeben werden.

Desk Daten Wahl Rechnen Schalter Optionen Fonts Programme
ADRESSEN.Namensschlüssel

Namensschlüssel▷ D..Rauterberg, Matthias.....

Anrede: Herrn.....
Vorname: Matthias..... Nachname: Rauterberg.....
Straße: Kurvenstraße 1.....

Ort: 8006 Zürich..... Land: Schweiz.....

Tel: -.....

Res.-Wunsch erh.: ___/___/___ Anreise: ___/___/___ Abreise: ___/___/___
Platz-Angebot▷ ___ Platz-Angebot abges.: ___/___/___
Res.-Bestät. erh.: ___/___/___ Res.-Betrag erh.: ___/___/___

Bemerkung: war noch nie auf diesem Campingplatz.....

VOR ABBRUCH ZURÜCK

EINGEBEN LÖSCHEN ÄNDERN SUCHEN ANZEIGEN DATEI SCHLÜSSEL DATEIINF DE INFO ENDE MOHT 1

Abbildung 5: Ausgabe eines Datensatzes als Maske in einer Dialogbox. Die weitere Dialogsteuerung kann nun nur noch über einen der vorhandenen Dialogknöpfe ("VOR", "ABBRUCH" oder "ZURÜCK") erfolgen.

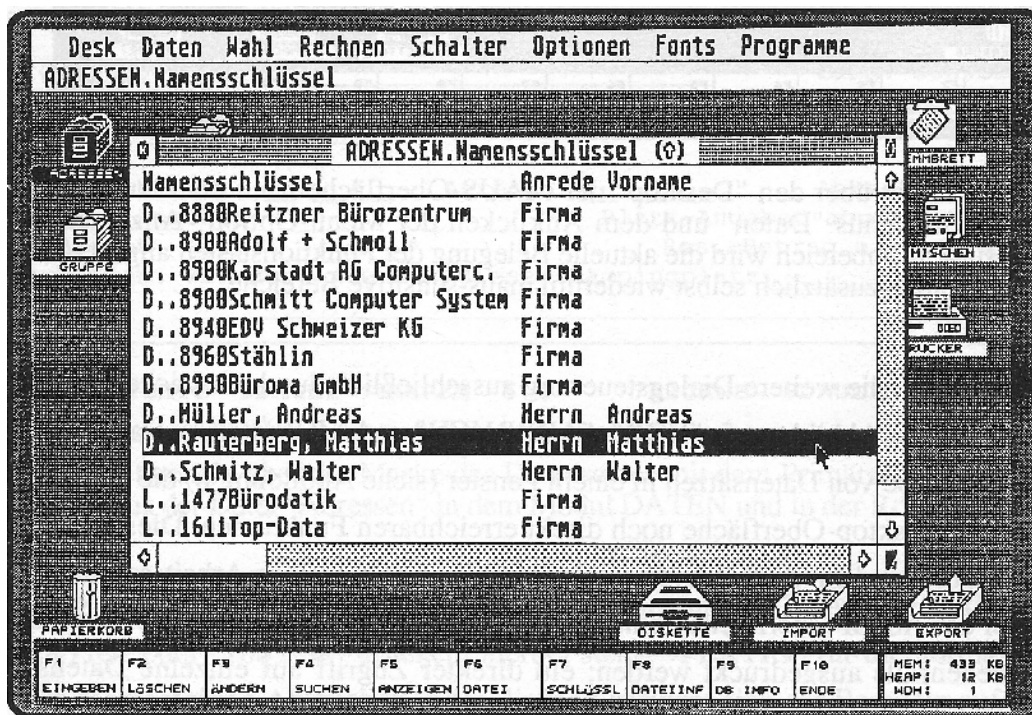


Abbildung 6: Ausgabe von Datensätzen als Liste in einem Fenster. Der durch einen einfachen Mausklick aktivierte Datensatz kann über die entsprechenden Menü-Optionen oder über eines der Icons auf dem Desktop weiter verarbeitet werden.

Der Benutzer kann nun durch einen Doppel-Klick auf eine Zeile, die den gesuchten Datensatz enthält, den Inhalt dieses Datensatzes in einer Dialogbox als Maske erscheinen lassen. Wird diese Zeile jedoch nur durch einen Einfach-Klick aktiviert (inverse Darstellung; siehe Abbildung 6), so kann der Benutzer folgende Operationen für diesen Datensatz auslösen: Löschen, Korrigieren, bzw. durch Verschieben mit der Maus auf eines der vorhandenen Ikons die jeweilige Funktion (Klemmbrett, Drucken, Export, Mischen, Löschung in den "Papierkorb").

Wie aus den vorherigen Abbildungen ersichtlich wurde, hat der Benutzer bei beiden Oberflächen im Rahmen dieser Untersuchung die gleiche Funktionalität zur Verfügung. Der Unterschied der beiden Oberflächen besteht jedoch in der unterschiedlichen Art der Dialogsteuerung (MAUS versus FUNKTIONSTASTE). Während der Benutzer sich bei der ASCII-Oberfläche nur innerhalb der aktuellen Menü-Hierarchie bewegen kann, ist der Zugriff auf die einzelnen Funktionen bei der MAUS-Oberfläche direkter und flexibler.

Folgende zwei Fragen sollen nun mit dieser Untersuchung beantwortet werden:

- 1.) Gibt es einen arbeitswissenschaftlich relevanten Unterschied in der Bearbeitungszeit zwischen diesen beiden Oberflächen bei der Handhabung eines relationalen DBMS ?
- 2.) Gibt es einen Zusammenhang zwischen der Art der Aufgabe und dem Typ Benutzungsoberfläche ?

4. Methodisches Vorgehen

Es ergibt sich ein zweifaktorielles, varianzanalytisches Design mit Meßwiederholung auf einem Faktor: der erste Faktor ist der "Typ der Benutzungsoberfläche" ("ASCII" auf IBM unter MS-DOS vs. "MAUS" auf IBM unter GEM) und den zweite Faktor bilden die 10 "Standard-Aufgaben". Diese zwei Faktoren sind die beiden unabhängigen Variablen. Es ergeben sich somit zwei eigenständige Benutzer-Gruppen (2 * 6 Benutzer = 12 N).

Als abhängige Variablen wurden erhoben: die reinen Bearbeitungszeiten gemäß Logfile-Protokoll (bereinigt von den System-Antwortzeiten; USERTIME), die mittlere Reaktionszeit pro Tastendruck (TIME-LEVEL) und die Anzahl der Online-Hilfefunktionsaufrufe; als Kontrollvariablen: die genaue Anzahl der Stunden an allgemeiner EDV-Vorerfahrung und die genaue Anzahl der Stunden an spezifischer Vorerfahrung mit der jeweiligen Benutzungsoberfläche.

Die Zeitmessung der Variablen USERTIME erfaßt ausschliesslich diejenige Zeitspanne bei jedem Tastendruck, welche der Benutzer nach Beendigung der Systemausgabe bis zu seiner nächsten Eingabe benötigt; es bleiben also alle zeitlichen Anteile, welche alleine durch die Systembearbeitung (wie z.B. Ausgabe von Datensätzen, Ausgabe von Listen, etc.) zustande kommen, unberücksichtigt.

Für die Bearbeitung der 10 Standard-Aufgaben stand das Datenbankprogramm ADIMENS mit den zwei ausgewählten Benutzungsoberflächen zur Verfügung, wobei als Anwendungskomponente exakt die gleiche Datenbankmaschine diente; der Zeitpunkt und die Bedeutung eines jeden Tastendruckes wurden automatisch in einem Logfile protokolliert.

4.1. Beschreibung der Benutzer

Es nahmen 12 Experten aufgeteilt in zwei Gruppen an dieser Studie teil. Diese 12 Experten zeichneten sich dadurch aus, daß sie in ihrer täglichen Arbeit mit dem DBMS schon seit mehreren Jahren mit dem jeweiligen Typ an Oberfläche gearbeitet haben. Die Experten erhielten keine Bezahlung.

Gruppe-1 (Experte-Ascii): durchschnittlich 38 Jahre; 6 Männer; 7.500 Std. allgemeine EDV-Vorerfahrung; 1.736 Std. spezifische Oberflächenerfahrung.

Gruppe-2 (Experte-Maus): durchschnittlich 38 Jahre; 6 Männer; 3.700 Std. allgemeine EDV-Vorerfahrung; 1.496 Std. spezifische Oberflächenerfahrung.

4.2. Ablauf der Untersuchung

Die Experten begannen nach der Erhebung ihrer Vorerfahrung durch einen Vorerfahrungsfragebogens (16 Skalen; insgesamt 115 Fragen) mit der Aufgabenbearbeitung. Am Ende füllten alle Benutzer einen Nachbefragungsbogen aus. Die Untersuchungen dauerten pro Benutzer 180 bis 240 Minuten (Einzelsitzungen).

Die Experten haben alle (bis auf einen) die 10 Aufgaben bearbeiten können. Die Reihenfolge der gestellten Aufgaben war für alle gleich. Erst wenn die jeweilige Aufgabe vollständig gelöst worden war, durften die Versuchspartner weiterarbeiten.

4.3. Beschreibung der zehn Standard-Aufgaben

Die zehn Aufgaben wurden so ausgewählt, daß exakt die gleiche Funktionalität der Anwendungskomponente des DBMS unter den beiden Oberflächen angesprochen werden konnte und die in der alltäglichen Arbeit am häufigsten vorkommenden Handlungsschritte durchgeführt wurden. Aufgabe 9 & 10 wurden ausgewählt, um die aufgabenangemessene Oberflächengestaltung zu testen. Die genaue Beschreibung dieser 10 Benchmark-Aufgaben ist bei RAUTERBERG (1989b) dargestellt.

Als Test-Datenbank diente eine Datenbank bestehend aus drei Dateien (PLATZ; 17 Datensätze, ADRESSEN; 280 Datensätze, GRUPPE; 27 Datensätze) zur Verwaltung eines Campingplatzes.

Aufgabe 1: Aktivieren einer bestimmten Menü-Option und Ablesen der Dateigrößen.

Konzept: hier mußten die Benutzer herausfinden, wie viele Datensätze in jeder der drei Dateien der Datenbank vorhanden sind; dazu gab es im Prinzip vier Lösungsmöglichkeiten.

Lösungen: "von Hand"; "Anzeigen als Liste"; "Gefundene Datensätze anzeigen."; "Datei-Info".

Bemerkung diese Aufgabe war als Aufwärm-Aufgabe gedacht. Bei beiden Oberflächen gibt es die Möglichkeit, sich diese benötigte Information über einen speziellen Transparenz-Operator ("Datei-Info") auf dem Bildschirm ausgeben zu lassen.

Aufgabe 2: Öffnen (sortiert nach einem vorgegebenen Schlüsselmerkmal), Selektieren und Löschen des letzten Datensatzes (für Datei: PLATZ, ADRESSEN, GRUPPE).

Konzept: in jeder der drei Dateien sollte der jeweils nach einem bestimmten Schlüsselmerkmal sortierte letzte Datensatz gefunden und dann gelöscht werden; hierzu gab es drei Lösungsansätze.

Lösungen: "von Hand"; "Sortierrichtung umdrehen"; "Fenster & Papierkorb".

Bemerkung diese Aufgabe war dazu gedacht, die sehr effiziente Möglichkeit der ASCII-Oberfläche, den letzten Datensatz einer Datei zu erhalten, der etwas anderen Methode der MAUS-Oberfläche gegenüberzustellen. Bei der ASCII-Oberfläche reichten in der Routine "Löschen" zwei Tastendrucke (Funktionstaste, RETURN) aus, während bei der MAUS-Oberfläche vorbereitend über einen Schalter die Sortierichtung umgekehrt werden mußte, bevor man die entsprechende Datei öffnen konnte. Dieser Lösungsansatz erfordert bei beiden Oberflächen von den Benutzern ein relativ hohes, abstraktes Vorstellungsvermögen über den Aufbau einer Datei. Bei der MAUS-Oberfläche gibt es dagegen die sehr anschauliche Möglichkeit, sich die einzelnen Datensätze als Inhalt einer Datei in sequentieller Form anzeigen zulassen ("Ausgabe als Liste"), wodurch die Eigenschaft eines Datensatzes "der letzte" visuell dem Benutzer unmittelbar transparent gemacht werden kann.

Aufgabe 3: Selektion eines bestimmten Datensatzes (Datei: PLATZ), Korrektur des Datensatzes bei 4 Merkmalen.

Konzept: hier mußte ein bestimmter Datensatz in einer angegebenen Datei gefunden und anschliessend nach vorgegebenen Angaben korrigiert werden; es gab hierfür vier Lösungsmöglichkeiten.

Lösungen: "zu Fuß"; "Korrektur mit SELEKTION"; "Korrektur mit WAHL"; "Korrektur über FENSTER".

Bemerkung Wie sich in der Vorstudie (RAUTERBERG, 1987) gezeigt hatte, sind die reinen Editiermöglichkeiten auf einem Datensatz bei der MAUS-Oberfläche zum Teil sehr umständlich. Diese Probleme sind jedoch gegenüber der recht umständlichen und zum Teil inkonsistenten Syntax bei der ASCII-Oberfläche vergleichsweise gering.

Aufgabe 4: Selektion einer Menge von Datensätzen (Datei: ADRESSEN), Korrektur jedes Datensatzes in einem Merkmal.

Konzept: es sollten alle Datensätze in einer vorgegebenen Datei mit einem bestimmten Merkmal (insgesamt 4 Datensätze) gefunden und anschliessend korrigiert werden; es gab vier Lösungsansätze.

Lösungen: "zu Fuß"; "Korrektur mit SELEKTION"; "Korrektur mit WAHL"; "Korrektur über FENSTER".

Bemerkung Während Aufgabe 3 den Schwerpunkt auf die Editierung eines einzelnen Datensatzes legte, wurde in dieser Aufgabe die Funktionalität der "Wahl" und gegebenenfalls die Funktionalität des "Globalen Ändern" getestet. Da die Anzahl der zu bearbeitenden Datensätze vergleichsweise klein war, konnte die Korrektur dieser Datensätze auch noch einzeln "von Hand" erfolgen.

Aufgabe 5: Definition eines Filters für ein Merkmal (Datei: PLATZ), Anwenden des Filters auf die Datei ; Ausgabe der gefundenen Datensätze auf dem Bildschirm.

Konzept: hier mußte ein Wahl-Templat definiert werden, mit dem dann alle dieser Wahl entsprechenden Datensätze auf den Bildschirm ausgegeben werden sollten; hier gab es nur eine Möglichkeit, diese Aufgabe zu lösen.

Lösung: "Wahl.Definition".

Bemerkung Obwohl schon bei den vorherigen Aufgaben die Funktionalität der "Wahl" verwendet werden konnte, wurde bei dieser Aufgabe explizit die Verwendung dieser Funktionalität gefordert. Dies zeigte sich als sinnvoll, da es bei den ersten 4 Aufgaben auch Lösungsmöglichkeiten gab, die ohne die "Wahl" ausgekommen sind, und einige Experten noch nie oder nur sehr selten mit der "Wahl" gearbeitet hatten.

Aufgabe 6: Laden eines Rechen-Programmes (Datei: PLATZ), Anwenden der Rechnung auf alle Datensätze, Ausgabe auf Bildschirm und Abspeichern der Ergebnisse.

Konzept: es sollte ein vorgefertigtes Rechen-Templat geladen werden und dann die dort definierte Rechnung auf alle Datensätze einer angegebenen Datei angewendet werden; da die Rechenergebnisse jedoch zunächst nur für die Ausgabe auf dem Bildschirm errechnet wurden, sollten die Rechenergebnisse auch noch in eine zu selektionierende Menge von Datensätzen abgespeichert werden; es gab drei Möglichkeiten diese Aufgabe korrekt zu bearbeiten.

Lösungen: "zu Fuß ohne WAHL"; "zu Fuß mit WAHL"; "Globales Ändern".

Bemerkung Bei dieser Aufgabe mußte nun die Funktionalität "Rechnen" benutzt werden. Die besondere Schwierigkeit dieser Aufgabe liegt darin, daß dem Benutzer nicht ohne weiteres transparent wird,

daß die über das geladene Rechentemplat errechneten Werte zunächst nur für die Ausgabe der Feldinhalte auf dem Bildschirm errechnet werden und nicht schon zum festen Bestandteil des Datensatzinhaltes gehören. Dieser Übergang von einer virtuellen zu einer realen Merkmalsausprägung mußte anschliessend über die "Wahl" nur für diejenigen Datensätze, die der vorgegeben Bedingung genügte, bewerkstelligt werden.

Aufgabe 7: Selektion einer Menge von Datensätzen (Datei: GRUPPE), Erstellen und Drucken einer Liste mit 3 Merkmalen für die gefundenen Datensätze.

Konzept: zunächst mußten nach einem vorgegebenen Kriterium die entsprechenden Datensätze einer Datei selektiert werden; dann sollte für diese Datensätze eine Liste erstellt werden, die anschliessend auszudrucken war; es gab drei Lösungsansätze.

Lösungen: "zu Fuß mit WAHL"; "direkt mit WAHL"; "Anzeigen als Liste mit WAHL".

Bemerkung Bei dieser Aufgabe sollte vorbereitend für die Aufgaben 9 und 10 das Erstellen einer Liste bezüglich nur einer Datei nach einem vorgegebenen Selektionskriterium getestet werden. Die Möglichkeit eine Liste zu erstellen wird bei der MAUS-Oberfläche durch ein eigenes Programm-Modul ("Anzeigen als Liste") besonders unterstützt, wohingegen bei der ASCII-Oberfläche die Listenerstellung über eine Reihe von Schaltern gesteuert wird.

Aufgabe 8: Suchen eines (nicht vorhandenen) Datensatzes (Datei: ADRESSEN), Selektion eines Datensatzes (Datei: PLATZ), Laden des Rechen-Programmes, Mischen mit dem Datensatz, Ausdrucken der erstellten Rechnung.

Konzept: es sollte zunächst in einer bestimmten Datei ein Datensatz nach einem vorgegebenen Kriterium gesucht werden; dieser Datensatz war jedoch nicht zu finden, weil es diesen Datensatz dort nicht gab; dann sollte nach dem gleichen Kriterium in einer anderen Datei gesucht werden und für den dort vorhandene Datensatz eine Rechnung erstellt werden; dazu mußte ein angegebenes Misch-Dokument mit diesem Datensatz zusammengemischt werden; dann war diese Rechnung noch auf den Drucker auszudrucken; es gab dabei im Prinzip zwei Bearbeitungswege.

Lösungen: "zu Fuß"; "direkt mit WAHL".

Bemerkung Durch diese Aufgabe sollte die Funktionalität des "Mischens" bei den beiden Oberflächen gegenübergestellt werden. Zusätzlich stellte sich als sehr aufschlußreich heraus, daß das Suchen eines nicht vorhandenen Datensatzes mit enormen Schwierigkeiten verbunden zu sein scheint.

Aufgabe 9: Selektion einer Menge von Datensätzen (Datei: GRUPPE), Erstellen und Drucken einer Liste mit 5 Merkmalen aus Datei PLATZ und GRUPPE (2-stellige Relation).

Konzept: es mußte eine Liste erstellt werden, die sich aus Datensätzen aus zwei verschiedenen Dateien zusammensetzte; dazu mußte mit dem Texteditor bei der Desktop-Version ein Misch-Dokument (bzw. mit dem Modul LISTE bei der Ascii-Version) erstellt werden, welches die Funktion einer Relation für diese beiden Dateien hatte; diese Liste sollte nur eine vorgegebene Auswahl an Merkmalen enthalten; vorher mußten die entsprechenden Datensätze nach einem vorgegebenen Kriterium aus einer der Dateien ausgewählt werden; anschliessend sollte diese Liste ausgedruckt werden; es gab nur eine Lösungsmöglichkeit.

Lösung: MAUS-Oberfläche

"Liste über 1stWORD+": zunächst wird über das Menü "Programme" mit der Option "Texteditor" das Textverarbeitungsprogramm 1stWORD+ gestartet und ein Dokument mit folgendem Inhalt in der Directory "\gemapps\db-camp" (auf Anweisung des Versuchsleiters wurden die einzelnen Textdateien fortlaufend mit L1...n benannt) angelegt:

```
#Platz:4# #Name:15# #Geb.Datum:8###Staatsangeh.:4##Pass-Nr:9#
#Namensschlüssel->Gruppe.Ansprechpartner#
#Nachname, Vorname:20# #Geb.-Datum:8# #Staatsangeh:4#
#<-#
```

nach Verlassen des Texteditors muß die DB-Camping "CAMP.INX" neu geladen werden; es wird die Datei PLATZ aktiviert, weil hier der Verbindungsschlüssel eindeutig ist; dann wird im Menü "Wahl" über die Option "Definieren" die folgende Wahl für das Merkmal "abgereist" (abgereist = "00/00/00") definiert und der Schalter "Verwenden" gesetzt; daran anschliessend wird das Datei-Ikon PLATZ auf das MISCH-Icon geschoben und das erstellte Dokument (z.B. L1.DOC)

dazugeladen; nachdem der Misch-Vorgang beendet ist, wird das MISCH-Icon auf das DRUCKER-Icon geschoben und die Liste ausgedruckt.

Lösung: ASCII-Oberfläche

"MODUL: Liste": es wird zuerst die Datei PLATZ aktiviert, weil hier der Verbindungsschlüssel eindeutig ist; dann wird im Modul "Wahl" über die Option "Definieren" die folgende Wahl für das Merkmal "abgereist" (abgereist = "00/00/00") definiert und der Schalter "Verwenden" gesetzt, bzw. die entsprechende Abfrage bejaht; dann wird über das Hauptmenü das Modul-"Liste" aufgerufen und interaktiv in der Routine "Definition" über die Funktionstaste "Marke" und interaktivem Verzweigen folgende Listen-Definition mit Hilfe der Funktionstaste "4MARKE" angelegt:

"Platz" "Name" "Geb.Datum" "Staatsangeh." "Pass-Nr"

nun wird interaktiv über das Merkmal "Namensschlüssel" in die Datei GRUPPE (Funktionstaste "3DATEI") verzweigt und die folgenden Merkmalsnamen mit "4MARKE" markiert: "Nachname, Vorname" "Geb.-Datum" "Staatsangeh"

und wieder in die Datei PLATZ über das Merkmal "Ansprechpartner" zurückverzweigt (Funktionstaste "9FERTIG");

nach dieser Definition wird im MODUL-Liste über die Routine "Erstellen" die gewünschte Liste auf dem Bildschirm ausgegeben; falls sie nicht den gewünschten Anforderungen entsprach, mußte die Definition erneut von vorne angefangen werden; daran anschliessend wird der Schalter für das Ausgabegerät auf DRUCKER gesetzt und im MODUL-Liste die Liste erneut erstellt, so daß sie ausgedruckt wird.

Bemerkung Bei der ASCII-Oberfläche wird die Funktionalität "Liste erstellen" durch ein recht mächtiges Modul unterstützt, bei dem interaktiv in die anderen Dateien einer Datenbank verzweigt und die dort in Frage kommenden Merkmale interaktiv ausgewählt werden können. Hingegen bei der MAUS-Oberfläche wird für die Erstellung von Listen mit Merkmalen aus mehreren Dateien der Aufruf eines ADIMENS fremden Texteditors notwendig, in dem dann eine Datei mit Steueranweisungen gemäß der definierten Misch-Syntax erzeugt werden muß. Dieser recht umständliche Weg bei der MAUS-Oberfläche im Gegensatz zur integrierten Version bei der ASCII-Oberfläche bedeutet die Frage nach der Aufgabenangemessenheit bei der Gestaltung von Oberflächen. Zusätzlich besteht hier die Schwierigkeit in der Definition des Selektionskriteriums.

Aufgabe 10: Selektion einer Menge von Datensätzen (Datei: PLATZ & ADRESSEN), Erstellen und Drucken einer Liste mit 3 Merkmalen aus Datei PLATZ, ADRESSEN und GRUPPE (3-stellige Relation).

Konzept: es mußte eine Liste erstellt werden, die sich aus Merkmalsausprägungen von Datensätzen aus drei verschiedenen Dateien zusammensetzte; dazu mußte mit dem Texteditor bei der Desktop-Version ein Misch-Dokument (bzw. mit dem Modul LISTE bei der Ascii-Version) erstellt werden, welches die Funktion einer Relation für diese drei Dateien hatte; diese Liste sollte nur eine vorgegebene Auswahl an Merkmalen enthalten; vorher mußten die entsprechenden Datensätze nach einem vorgegebenen Kriterium aus einer der Dateien ausgewählt werden; anschliessend sollte diese Liste ausgedruckt werden; es gab nur eine Lösungsmöglichkeit.

Lösung: ähnlich wie bei Aufgabe 9.

Bemerkung Diese Aufgabe ist eine komplexere Variante der 9. Aufgabe, weil in dieser Aufgabe eine Relation über drei Dateien hinweg definiert werden mußte. Die meisten Schwierigkeiten bereitete die Wahl der Ausgangsdatei "Platz" und dann das Finden der richtigen Reihenfolge der Verzweigungen. Es hat sich bei der ASCII-Oberfläche als sehr nachteilig herausgestellt, daß bei dem kleinsten Versehen, die gesamte Liste jedesmal wieder neu von ganz vorne interaktiv definiert werden mußte.

Die Desktop-Benutzer mußten für Aufgabe 9 und 10 ein Mischdokument mit einem externen Texteditor nach der Syntax einer vorgegebenen, einfachen Kommandosprache erstellen, während die Ascii-Benutzer die Listen interaktiv in einem Listen-Modul definieren und erstellen konnten. Diese beiden Aufgaben wurden mit in diese Untersuchung hineingenommen, um das Kriterium der "Aufgabenangemessenheit" mit untersuchen zu können (ULICH, 1987).

5. Darstellung der Ergebnisse

Es wurde ein zwei-faktorielles Design über alle zehn Aufgaben hinweg gerechnet (Faktor "Aufgabe (1-10)" und Faktor "Oberfläche").

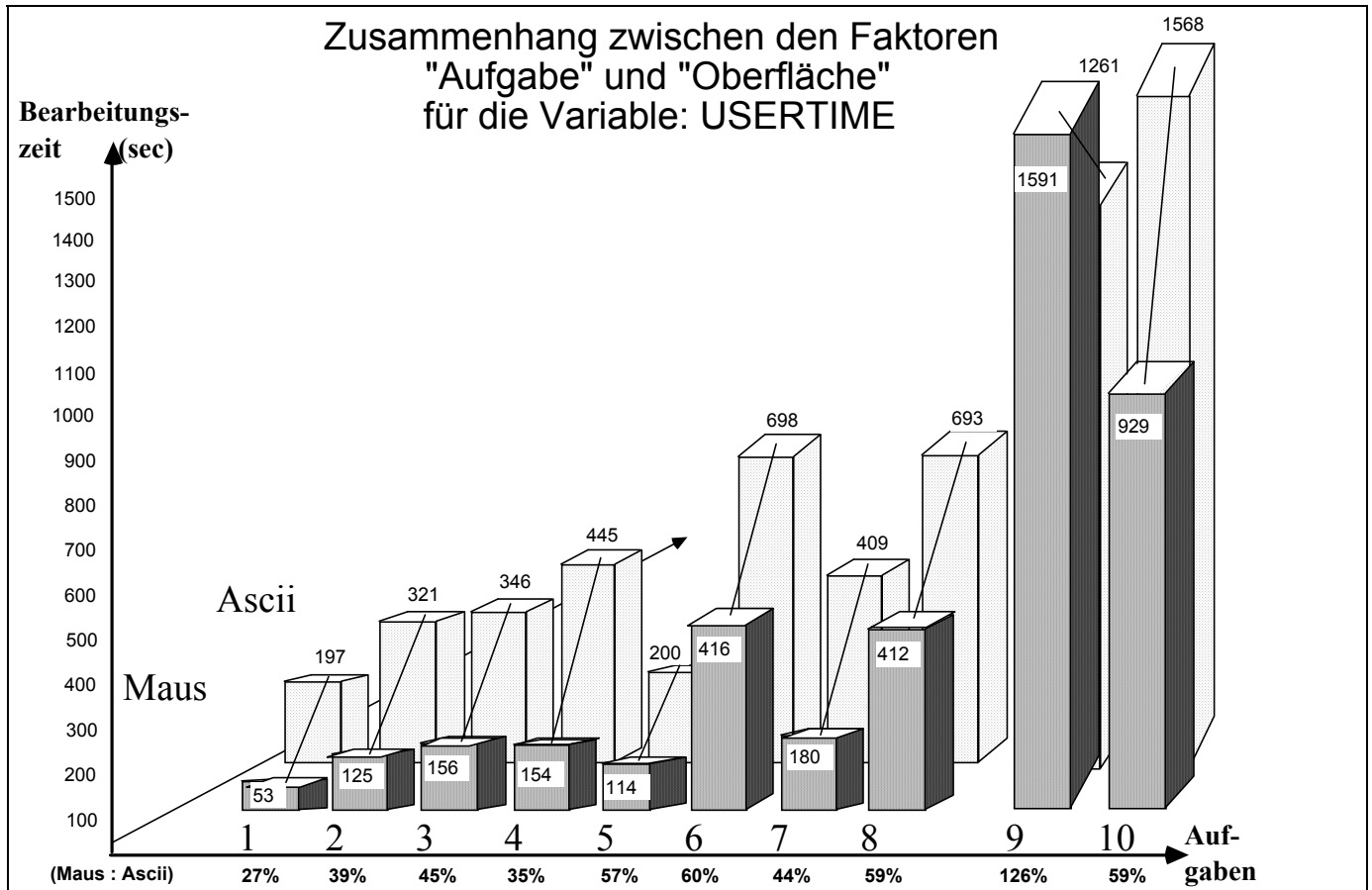


Abbildung 7: Darstellung der gemessenen, reinen Bearbeitungszeiten (Variable USERTIME) für die einzelnen Aufgaben, getrennt nach Oberfläche: "Maus" = direkt-manipulative, bzw. "Ascii" = menü-orientierte Benutzungsoberfläche. In der untersten Zeile ist das prozentuale Verhältnis der Zeiten (Maus:Ascii) angegeben.

Im Mittel reichte für die Experten mit der direkt-manipulativen Oberfläche ("Maus") nur 55% derjenigen Bearbeitungszeit aus, die die Benutzer mit der menü-orientierten Oberfläche ("Ascii") benötigten (USERTIME: $p=0.001^1$; siehe Tabelle 1); läßt man die Aufgaben 9 und 10 außen vor, so verbessert sich das Verhältnis sogar auf durchschnittlich 46%. Wie erwartet, kehrt sich bei Aufgabe 9 das Verhältnis zugunsten der menü-orientierten Oberfläche um (126%, Abbildung 7; $p=0.002 - 0.003$, Tabelle 1).

¹dieser p-Wert gibt mit 100% multipliziert die Fehlerwahrscheinlichkeit (α -Fehler) für die Aussage an, daß ein Unterschied zwischen den beiden Oberflächen hinsichtlich der jeweiligen abhängigen Variablen besteht. Allgemein wird ein α -Fehler zwischen 10% und 5% oder kleiner als statistisch bedeutsam angesehen.

Es ergibt sich insgesamt kein Unterschied für die Reaktionszeit pro Tastendruck (TIMELEVEL: $p=0.312 - 0.897$, siehe Tabelle 1). Besonders auffällig sind jedoch die relativ langen Reaktionszeiten der "Ascii"-Benutzer bei Aufgabe 1 (Ascii:Maus = 9.1 sec : 4.0 sec = 228%) und bei Aufgabe 3 (Ascii:Maus = 2.2 sec : 1.3 sec = 169%), sowie umgekehrt der "Maus"-Benutzer bei Aufgabe 4 (Maus:Ascii = 4.5 sec : 3.0 sec = 150%). Diese Unterschiede in dem Zusammenhang der Faktoren "Aufgabe" und "Oberfläche" sind jedoch nur tendenziell bedeutsam ($p=0.086 - 0.126$; siehe Tabelle 1).

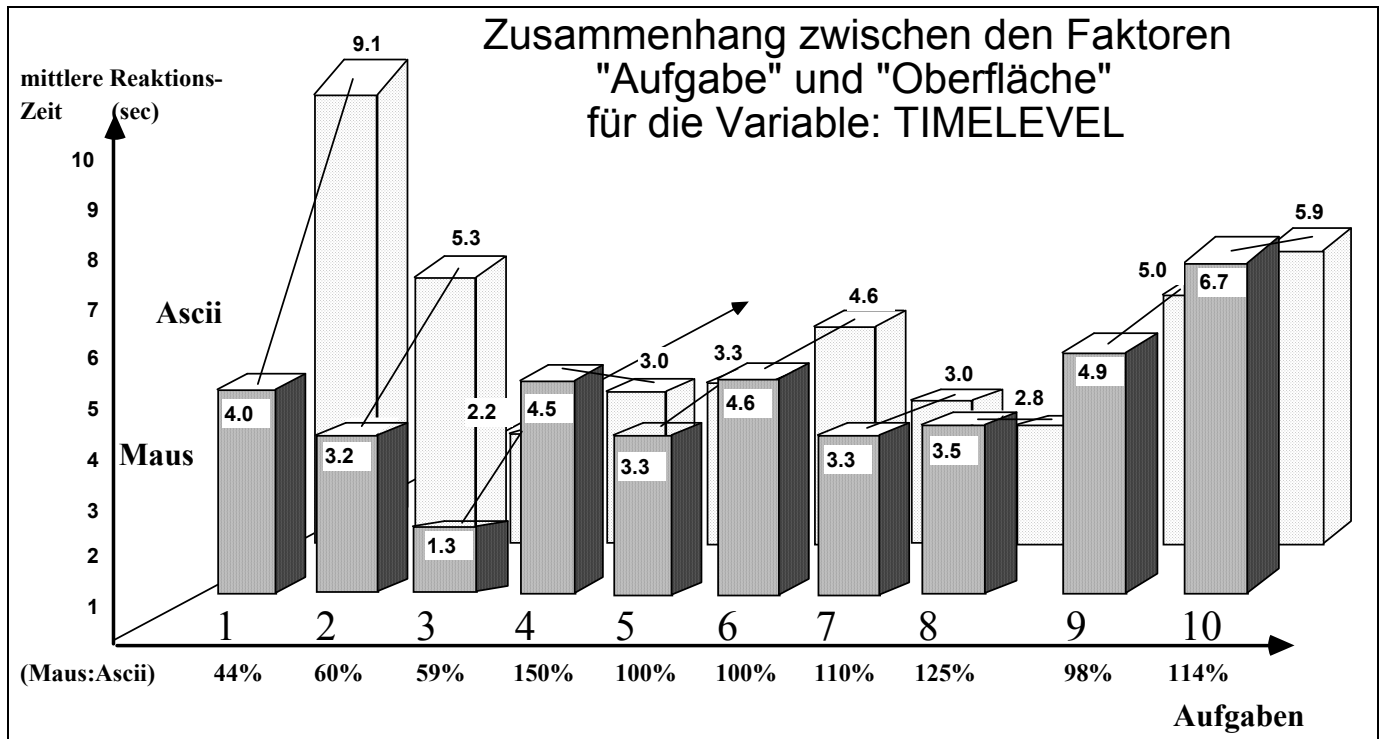


Abbildung 8: Darstellung der mittleren Reaktionszeit pro Tastendruck (Variable TIMELEVEL) für die einzelnen Aufgaben, getrennt nach Oberfläche: "Maus" = direkt-manipulative, bzw. "Ascii" = menü-orientierte Benutzungsoberfläche. In der untersten Zeile ist das prozentuale Verhältnis der Reaktionszeiten (Maus:Ascii) angegeben.

Zieht man nun den bedeutsamen Unterschied hinsichtlich des Faktors "Oberfläche" für die Variable USERTIME bei der Interpretation des nicht signifikanten Ergebnisses für die Variable TIMELEVEL heran, so muß man zu dem Schluß kommen, daß die Benutzer der ASCII-Oberfläche entsprechend mehr Tastendrucke bei der Bearbeitung der einzelnen Aufgaben benötigen. In der Tat ergibt nun eine entsprechende varianzanalytische Auswertung der Anzahl der benötigten Tastendrucke ein entsprechend signifikantes Ergebnis ($p=0.001$).

Tabelle 1: Aufgaben (1-10) (p =Signifikanzniveau)	Kontroll-Variablen: allgemeine EDV-Erfahrung		Kontroll-Variablen: spez. Oberflächenerfahrung	
	USERTIME	TIMELEVEL	USERTIME	TIMELEVEL
Aufgabe	0.001	0.001	0.001	0.001
Oberfläche	0.001	0.897	0.001	0.312
Aufgabe \square Oberfläche	0.002	0.086	0.003	0.126

Gewinner in diesem "Rennen" ist die Gruppe 1 (Experten mit der direkt-manipulativen Oberfläche: mittlere Bearbeitungszeit über alle Aufgaben hinweg = 374 sec.); Verlierer ist Gruppe 2 (Experten mit menü-orientierter Oberfläche: mittlere Bearbeitungszeit über alle Aufgaben hinweg = 581 sec.).

Berechnet man den Korrelationskoeffizienten zwischen der Anzahl an Lösungsmöglichkeiten pro Aufgabe und dem prozentualen Verhältnis der Bearbeitungszeiten (Maus:Ascii; siehe Abbildung 7), so ergibt sich ein auf dem 5%-Niveau statistisch bedeutsamer Zusammenhang von $R = -0.69$; dieser sehr wichtige Zusammenhang bedeutet nun, daß die MAUS-Oberfläche insbesondere bei den Aufgaben mit einer Vielzahl an Lösungsmöglichkeiten deutlich von Vorteil ist (Aufgaben 1 bis 4 und 7).

6. Diskussion

Obwohl die Experten der ASCII-Oberfläche mehr als doppelt soviel an Vorerfahrung im Umgang mit EDV allgemein und mehr an spezifischer Vorerfahrung im Umgang mit ihrer DBMS-Oberfläche haben gegenüber ihren MAUS-Kollegen (siehe Abschnitt 4.1), läßt sich eine deutliche Überlegenheit der direkt-manipulativen Benutzungsoberfläche gegenüber der konventionellen, menü-orientierten Oberfläche aufzeigen. Frage 1 läßt sich daher eindeutig mit Ja beantworten. Mit der desktop-orientierten Oberfläche kann mehr als die Hälfte der Bearbeitungszeit eingespart werden. Dieser deutliche Unterschied in der Bearbeitungszeit kann nicht über die unterschiedliche Reaktionszeit pro Tastendruck erklärt werden, sondern dadurch, daß die Desktop-Benutzer entsprechend weniger Tastendrucke insgesamt für die Bearbeitung jeder Aufgabe benötigten. Dies wird auf die größere Transparenz der Desktop-Oberfläche zurückgeführt. Je besser die Oberfläche den Benutzer bei seiner Orientierung hinsichtlich des aktuellen Dialog- und Anwendungszustandes unterstützt, desto schneller und direkter kann der Benutzer seiner Aufgabebearbeitung nachgehen.

Dies läßt sich an den Ergebnissen für die beiden Aufgaben 1 und 2 verdeutlichen. Während die Desktop-Benutzer durch einen einfachen Mausklick auf das Feld "F8 DATEIINF" (siehe Abbildung 4 und 6) sich die benötigten Informationen für Aufgabe 1 auf den Bildschirm holen konnten, mußten die Benutzer der ASCII-Oberfläche erst die Routine "Datei" in dem Modul "Info" aktivieren. Während bei der Desktop-Oberfläche der Wechsel von einer Datei zu einer anderen lediglich durch einen einfachen Maus-Klick auf das entsprechende Datei-Ikon erfolgt, mußten die Benutzer der ASCII-Oberfläche jeweils auf der Modul-Ebene "Info" über die Funktionstaste "3DATEI" und Eingabe der Datei-Nummer den Dateiwechsel vornehmen.

Bei der Aufgabe 2 ("Löschen des letzten Datensatzes") war gegenüber der Aufgabe 1 eigentlich die ASCII-Oberfläche bevorzugt: die Benutzer konnten von der Hauptmenü-Ebene über die Eingabesequenz "D" für das Modul "Daten", "L" für die Routine "Löschen", "2RICHTG" für die Umkehrung der Sortierichtung, "RETURN" für die Ausgabe des letzten Datensatzes und "9FERTIG" den gesuchten Datensatz löschen. Es stellte sich jedoch heraus, daß die ASCII-Benutzer sehr häufig zwischen dem letzten und dem ersten Datensatz hin- und hergewechselt sind, um auch ja sicher zu sein, nur den letzten Datensatz zu löschen. Hier war die Desktop-Oberfläche mit der Möglichkeit, sich die Datensätze der Reihe nach sortiert in einem Fenster anzeigen zu lassen, überlegen. Die Benutzer ließen sich die ersten

Datensätze der jeweiligen Datei in einem Fenster ausgeben, verschoben den Rollbalken am rechten Fensterrand nach unten und transportierten den letzten Datensatz auf den Papierkorb. Hier erweist sich die Eigenschaft dieser Darstellungsart, nicht nur den jeweils gesuchten Datensatz, sondern auch seine Datensatz-Umgebung mit auszugeben, als besonders vorteilhaft.

Die Ausgabe der Datensatz-Umgebung erwies sich auch bei der Aufgabe 8 ("Suche nach einem nicht vorhandenen Datensatz,...") als wertvoll; nachdem die Versuche der Desktop-Benutzer, über die "Selektion", bzw. die "Wahl" stets zu einem negativen Rechercheergebnis führten - ebenso wie bei den Benutzern der ASCII-Oberfläche -, wurde über die Ausgabe als Liste die entsprechende Datensatz-Umgebung aufgesucht. Erst hierdurch konnten sich viele Desktop-Benutzer davon endgültig überzeugen, daß der gesuchte Datensatz nicht an der entsprechenden Stelle in der sortierten Liste vorhanden war.

Die Umstellung von einer menü-orientierten auf eine Desktop-Oberfläche alleine reicht jedoch nicht aus (Aufgabe 9 !): es muß für jeden Aufgabentyp eine sorgfältige, aufgabenangemessene Dialoggestaltung durchgeführt werden (Frage 2: Ja). Hatten die Experten der Desktop-Oberfläche jedoch erst einmal das Lösungsschema für die Aufgabe 9 (Definition einer Relation) herausgefunden, so konnten sie dies auch gleich gewinnbringend bei der Aufgabe 10 umsetzen (59% Bearbeitungszeit; siehe Abbildung 7). Dies spricht für die Lernförderlichkeit der direkt-manipulativen Benutzungsoberflächen.

Aus den Ergebnissen zu diesem Experiment lassen sich für die Desktop-Oberfläche eines DBMS vier wesentliche Problembereiche herausarbeiten:

1.Problem das Wechselspiel von "Selektion" und "Wahl":

Wie sich herausgestellt hat, kommt der Option "Anzeigen als Liste" eine zentrale Bedeutung zu. Für diese Anzeigeform konnte jedoch keine Selektion vorgeschaltet werden, so daß hierdurch die Schwierigkeiten in der "erzwungenen" Benutzung der "Wahl" als globalem Filter erklärt werden können (RAUTERBERG, 1987). Dieser Mechanismus ist jedoch nicht für diesen Zweck vorgesehen, so daß die meisten Schwierigkeiten im Umgang mit der Wahl sich durch die Möglichkeit der temporären Selektion vor der listenartigen Ausgabe der Datensätze in einem Fenster beheben ließen.

2.Problem die adäquate Rückmeldung von nicht-vorhandenen Datensätzen bei negativem Recherche-Ergebnis (siehe Aufgabe 8):

Wie sich bei Aufgabe 8 gezeigt hat, ist es offenbar für die Benutzer der Desktop-Oberfläche sehr hilfreich bei der Interpretation einer negativen Suchanfrage, daß die Benutzer die Möglichkeit hatten, sich die aktuelle Datensatzumgebung in Form einer Liste anzeigen lassen zu können, um dann daran zu erkennen, daß an der vorgesehenen Stelle der gesuchte Datensatz nicht vorhanden ist. Dies ließe sich vom System bei einem negativen Rechercheergebnis dahingehend unterstützen, daß jeweils die aktuelle Datensatzumgebung (z.B. +/- 5 Datensätze) als Liste unter einem entsprechenden Hinweis seitens des Systems ausgegeben wird.

3.Problem das Finden und Löschen des nach einem vorgegebenen Sortiermerkmal letzten Datensatzes einer Datei (siehe Aufgabe 2):

Ebenso wie bei dem vorherigen Problem zeigte es sich, daß die Ausgabe von Datensätzen einer Datei als sortierte Liste in einem Fenster bei der Bestimmung des letzten Datensatzes von großer Hilfe war. Auch hier geht es generell um das Problem bei der Suche nach einem bestimmten Datensatz, den genauen Standort dieses gesuchten Datensatzes visuell optimal umzusetzen.

4.Problem der fehlende Report-Generator bei der Desktop-Oberfläche zur Erstellung von Listen mit Merkmalen aus mehreren Datei einer Datenbank (siehe Aufgabe 9 und 10):

Bei der Bearbeitung der neunten Aufgabe zeigte sich bei den Desktop-Experten eine weitgehende Unkenntnis über die Möglichkeiten eine Relation zu definieren. Dies wird im wesentlichen darauf zurückgeführt, daß die angebotene Dialogstruktur mit dem Umweg über einen Texteditor zur Erstellung eines Misch-Dokumentes von den Experten als abschreckend erlebt wurde. Auf Nachfragen stellte sich heraus, daß die Generierung von druckreifen Ausgabelisten dann über die zugehörige Retrieval-Sprache ("ADI-Talk", dBASE ähnlich) abgewickelt wurde. Diese für ein relationales Datenbanksystem eigentlich kennzeichnende Eigenschaft - die Erstellung von Relationen über die verschiedenen Dateien einer Datenbank hinweg - war den meisten Desktop-Benutzern weitgehend fremd. Eine wesentliche Bereicherung stellt daher die Implementation eines komfortablen Report-Generators dar.

Insgesamt kann festgehalten werden, daß die direkt-manipulative Benutzungsoberfläche mit der "Maus" als generellem Interaktionselement der konventionellen, menü-orientierten Oberfläche mit den "Funktionstasten" deutlich überlegen ist.

Dies gilt insbesondere für Benutzer mit langer Vorerfahrung (Experten).

7. Literatur

- ALTMANN, A. (1987): Direkte Manipulation: empirische Befunde zum Einfluß der Benutzeroberfläche auf die Erlernbarkeit von Textsystemen. Zeitschrift für Arbeits- und Organisationspsychologie, 31 (N.F.3) 3, 108-114.
- HAUPTMANN, A.G. & GREEN, B.F. (1983): A comparison of command, menu-selection and natural-language computer programs. Behaviour and Information Technology, Vol. 2, No. 2, 163-178.
- HUTCHINS, E.L., HOLLAN, J.D. & NORMAN, D.A. (1986): Direct manipulation interfaces. In: User centered system design. (D.A. NORMAN & S.W. DRAPER, eds.), Hillsdale London: Lawrence Erlbaum; 87-124.
- KRAUSE, J. (1986): Direkte Manipulation elektronischer Objekte und Metaphernverwendung. LIR-Arbeitsbericht. Regensburg: Linguistische Informationswissenschaft.
- RAUTERBERG, M. (1987): Untersuchung der Benutzerfreundlichkeit einer desktop-orientierten Benutzungsoberfläche am Beispiel eines relationalen Datenbanksystems. Forschungsbericht, Zürich: ETH-Lehrstuhl für Arbeits- und Organisationspsychologie.
- RAUTERBERG, M. (1989): MAUS versus FUNKTIONSTASTE: ein empirischer Vergleich einer desktop- mit einer ascii-orientierten Benutzungsoberfläche. In: Software-Ergonomie '89. (MAASS, S. & OBERQUELLE, H.; Hrsg.). Stuttgart: Teubner; S. 313-323.
- RAUTERBERG, M. (1989b): Die Transparenz von desktop-orientierten im Vergleich zu konventionellen menü-orientierten Benutzungsoberflächen. Forschungsbericht, Zürich: ETH-Lehrstuhl für Arbeits- und Organisationspsychologie.
- ROBERTS, T.L. & MORAN T.P. (1983): The Evaluation of Text Editors: Methodology and Empirical Results. Communications of the ACM, Vol. 26, No. 4, 265-283.
- SHNEIDERMAN, B. (1987): Designing the user interface. Amsterdam Sydney Tokyo: Addison-Wesley.
- SMITH, S.L. & MOSIER, J.N. (1986): Guidelines for designing user interface software. MITRE Report No. AD/A177 198; Springfield, VA: NTIS, 1986.
- STREITZ, N.A.; SPIJKERS, W.A.C. & van DUREN, L.L. (1987): From Novice to expert user: a transfer of learning on different interaction modes. In: INTERACT '87. (BULLINGER, H.J. and SHACKEL, B., eds.), New York Oxford Tokyo: North-Holland; 841-846.
- ULICH, E. (1987): Some aspects of user-oriented dialogue design. In: System design for Human Development and Productivity: Participation and Beyond. (FUCHS-KITTOWSKI, K., DOCHERTY, P., KOLM, P. & MATHIASSEN, L., eds.) Amsterdam: North-Holland.
- WHITESIDE, J.; JONES, S.; LEVY, P.S.; WIXON, D. (1985): User Performance with Command, Menu, and Iconic Interfaces. Human Factors in Computing Systems-II. Proceedings of the CHI '85 Conference in San Francisco, Amsterdam New York Oxford: North-Holland; 185-191.