Interaktion : Dialog mit Werkzeugen des Informationszeitalters

Autor(en): **Kuhn, W.**

Objekttyp: **Article**

Zeitschrift: **Vermessung, Photogrammetrie, Kulturtechnik : VPK = Mensuration, photogrammétrie, génie rural**

Band (Jahr): **86 (1988)**

Heft 11

PDF erstellt am: **11.09.2024**

Persistenter Link: https://doi.org/10.5169/seals-233797

Interaktion: Dialog mit Werkzeugen des Informationszeitalters

W. Kuhn

Dies ist der dritte Beitrag in der Reihe der Nachdrucke von Vorträgen aus der Informationstagung vom 23./24. Oktober 1987 über die Ausbildung des Kulturingenieurs in Informatik im Vermessungswesen an der ETH Zürich. Er befasst sich mit der Interaktion zwischen Mensch und Computer und den Forschungs- und Entwicklungsarbeiten auf diesem Gebiet unter Professor Conzett. Weder in den Unterrichtsveranstaltungen noch in der Informationstagung konnte dieses vielseitige Thema systematisch behandelt werden. Dementsprechend soll auch dieser Artikel eher die Aufmerksamkeit auf einige Fragen rund um den Kontakt zwischen Mensch und Computer (speziell zwischen Ingenieur und LIS) lenken, als solche Fragen zu beantworten. Die zitierte Literatur ermöglicht eine gezielte Vertiefung in die eine ode andere Richtung. Als konkrete Anwendung und als Prototyp zu einem Forschungsprojekt über geometrisches Konstruieren wird das Macintosh-Programm HILS vorgestellt.

Cet article est le troisième dans la série issue des exposés lors des journées d'information sur la formation en informatique dans la mensuration des ingénieurs du génie rural à l'EPFZ, journées qui se sont tenues le 23/24 octobre 1987. Il traite de l'interaction entre l'utilisateur et l'ordinateur et des travaux de recherche et de développement réalisés dans ce domaine sous l'édige du professeur Conzett. Ni les cours pour les étudiants, ni les journées d'information n'ont permis de présenter ce thème pluridisciplinaire en profondeur. Ainsi, cet article vise plutôt à une sensibilisation pour quelques questions autour du contact entre utilisateur et ordinateur (surtout entre ingénieur et SIT) qu'à des réponses précises à ces questions. La littérature citée permet un approfondissement dans l'une ou l'autre direction. L'article se termine par la présentation du programme HILS sur Macintosh, comme application concrète et comme prototype accompagnant un projet de recherche sur la construction géometrique.

1. Was ist Interaktion?

1.1. Die Begriffe *«Interaktion»* und *«Benützerschnittstelle»*

Interaktion bedeutet «Wechselwirkung» zwischen oder «aufeinander einwirken» von Mensch und Computer. Dieser Kontakt zwischen Benützer und System erfolgt an der *Benützerschnittstelle,* einer der wichtigsten Komponenten eines Systems.

Mit der Benützerschnittstelle werden

- die Eingaben des Benützers in das System
- die Ausgaben des Systems an den Benützer
- die Reihenfolge dieser Ein- und Ausgaben

festgelegt.

Interaktion hat *physiologische und psychologische Aspekte:* Die physiologischen betreffen die Interaktions-Mittel (Tastatur, Bildschirm, Maus, Ton usw.) und die psychologischen die Vorstellung, die sich der Benützer vom System macht und seine Einstellung zum System.

In einem weiteren Rahmen hat Interaktion auch *gesellschaftliche Aspekte.* Der Computer kann als Werkzeug unserer Gesellschaft angesehen werden, das die soziale Situation der meisten Menschen verändert, auf erwünschte oder unerwünschte Art. Er beeinflusst die Gestaltung von Arbeitsplätzen und Haushalten, bringt neue Kommunikationsformen mit sich und beeinflusst damit indirekt die Interaktion zwischen Menschen.

1.2. Die «Human Factors» von EDV-Systemen

Als *Qualitätsmerkmale* für die Beurteilung von Benützerschnittstellen lassen sich die folgenden *quantifizierbaren* Faktoren angeben [Shneiderman 1987]:

- *Lernaufwand:* Wie lange brauchen die Benützer, um die Bedienung eines Systems zu lernen?
- *Arbeitsgeschwindigkeit:* Wie lange dauert die Bearbeitung typischer Aufgaben mit dem System?
- *Fehlerrate:* Wie viele Fehler welcher Art werden beim Arbeiten mit dem System gemacht?
- *Befriedigung:* Wie sind die Benützer subjektiv mit dem System zufrieden?
- *Einprägsamkeit:* Wie gut behalten die Benützer ihre Kenntnisse über die Systembedienung?

Natürlich sind diese sogenannten *«human factors»* nur allgemeine Kriterien, die oft auch gegeneinander abzuwägen sind: Längere Einarbeitungszeit ermöglicht höhere Arbeitsgeschwindigkeit, Fehlerraten können durch langsameres Arbeiten gesenkt werden usw. Je nach Anwendungsgebiet kommt einzelnen Faktoren mehr Gewicht zu als anderen: Bei Systemen zur Steuerung gefährlicher Anlagen spielt etwa die Fehlerrate sicher eine grössere Rolle als der Lernaufwand.

Sind «human factors» aber überhaupt wichtig? Lohnt es sich, bei der Entwicklung und beim Kauf von Systemen darauf zu achten?

2. Warum ist Interaktion ein Problem?

«Geringere Kosten ermöglichen es den Leuten, Computer zu *kaufen,* aber erst bessere Benützerschnittstellen ermöglichen es ihnen, Computer zu *benützen.»* [Smith et. al. 1982]

2.1. Der Computer verlässt die Rechenzentren

Die EDV-Umfrage 1987 der Informatikkommission des SVVK hat ergeben, dass in über 90% der befragten Kulturtechnik- und Vermessungsbüros häufig oder viel mit EDV gearbeitet wird und dass dies zunehmend auf büroeigenen Anlagen geschieht. Sei es bei der Datenerfassung im Feld [Brügger 1988], bei der Auswertung im Büro [Wigger 1988], bei der Arbeit an graphisch-interaktiven Systemen, bei der Planerstellung, aber auch bei der Büroadministration: der Geometer ist täglich konfrontiert, oder besser, in Interaktion mit EDV-Systemen.

Da der Computer heute in *fast alle Lebensbereiche* vordringt, genügt es nicht mehr, Software für EDV-Spezialisten zu schreiben. Auch EDV-Laien und Fachleute aus Anwendungsgebieten müssen damit umgehen können. Die Interaktion muss auch *verschiedenen Erfahrungshorizonten angepasst* werden können, da die Benützer bei der Arbeit mit Systemen lernen, und somit verschieden weit fortgeschrittene Benützer mit dem gleichen System arbeiten. Geübte möchten mehr Möglichkeiten und abgekürzte Methoden, Anfänger können damit aber überfordert werden.

Institut für Geodäsie und Photogrammetrie, ETH-Hönggerberg, CH-8093 Zürich
Separata Nr. 139.

2.2. EDV-Systeme als Werkzeuge

Seit den Anfängen der Zivilisation hat der Mensch für seine Tätigkeiten *Werkzeuge* entwickelt. Diese müssen einerseits ihren Zweck erfüllen, andererseits aber auch *dem Menschen angepasst* sein: Eine Schaufel mit einem Stiel viereckigen Querschnitts ist kaum viel nützlicher als eine, die beim zweiten Spatenstich bricht.

Entwickler von EDV-Systemen haben ihre Aufmerksamkeit lange nur auf die *Funktionalität* dieser Werkzeuge gerichtet, statt sich auch mit deren *Benützbarkeit* zu befassen. Dies hat, neben psychologischen und sozialen Auswirkungen, auch zu beträchtlichen wirtschaftlichen Einbussen geführt. Heute geht es ganz allgemein um die Frage, wie die *Werkzeuge des Informationszeitalters* ihren Benützern am besten dienen können.

2.3. Die Bedeutung der Interaktion

Die Qualität der Benützerschnittstelle entscheidet über den Erfolg oder Misserfolg eines Systems, also darüber, ob die Benützer das System *akzeptieren* und, wenn ja, ob sie damit *befriedigend und effizient arbeiten* können.

Ungeeignete Benützerschnittstellen können nicht nur höhere Schulungskosten und Fehlerraten verursachen, sondern auch dazu führen, dass ein System nach der Anschaffung gar nicht eingesetzt wird, oder dass die Benützer nur wenige Befehle lernen und das System nie voll ausnützen können (siehe z.B. die Fallstudien in [Bernet 1986]). Daraus ergibt sich die *wirtschaftliche,* aber auch die *psychologische* und *soziale Bedeutung* einer sorgfältigen Gestaltung der Interaktion.

Eine Untersuchung darüber, warum interaktive Systeme von potentiellen Anwendern oft nicht benützt werden, ergab hauptsächlich folgende Gründe [Nickerson 1981]:

- *Hoher «Eintrittspreis»:* Der Aufwand, um ein System zu starten und bis zum Punkt zu gelangen, wo man endlich seine Aufgabe bearbeiten kann, ist zu gross.
- *Mangelnde Funktionalität:* Das System wurde an den Bedürfnissen der Benützer «vorbeiprogrammiert».
- *Ungeeignete Befehlssprachen:* die Befehle sind schwierig zu lernen und zu behalten; sie bieten zuviele Fehlermöglichkeiten.
- *Inkonsistenz:* Verschiedene Programme verlangen ganz verschiedene Formen der Interaktion.
- *Schwache Leistung:* Das System ist zu langsam. Schlimmer: der Benützer weiss oft nicht, wie lange er auf eine Antwort warten muss, ja ob das System überhaupt noch arbeitet.
- *Unzuverlässigkeit:* Das System kann ohne Warnung zusammenbrechen, meist in dem Moment, wo der Schaden am grössten wird.
- *Überforderung:* Es fehlt die Zeit, die Systembenützung zu lernen.
- *Unverständlichkeit:* Der Benützer kann sich kein Bild machen, wie das System arbeitet. So wird er immer wieder überrascht oder frustriert durch unerwartete Reaktionen des Systems.
- *Mangelhafte Dokumentation:* Sie ist unklar, unvollständig, schlecht strukturiert oder veraltet.

Der folgende Dialog zwischen einem Programmierer und einem Benützer illustriert einige dieser Punkte: Der Benützer muss sich obskure Befehle merken, das Programm reagiert nicht oder auf unerwartete Weise, und kleine Fehler können fatale Wirkungen haben.

Programmierer: Nun, da du ein Stück des Schaltkreises gezeichnet hast, möchtest du vielleicht etwas daran ändern.
Benützer: Ja, löschen wir eine Komponente. Wie macht man das?
P: Zeig' auf das Menufeld CD.
B: CD?
P: Das steht für «Component Delete».
B: Aha. Also...he, was ist geschehen?
P: Du bist im Analysis Mode: du musst AM statt CD gewählt haben.
B: Seltsam, ich zeigte doch auf CD. Wie komme ich aus dem Analysis Mode heraus?
P: Gib Control-Q ein.
B. [schreibt C-O-N-T-R...]
P: Nein, halte die Control-Taste unten und drück' Q.
B: Sorry, dumm von mir...also, ich versuche CD nochmals.
P: Vielleicht zielst du ein bisschen über die Buchstaben, um nicht wieder in den Analysis Mode zu geraten – nein, nicht so hoch – so ist besser.
B: Geschafft!
P: Nun zeigst du auf die Komponente, um sie zu löschen.
B: Gut...es geschieht nichts; was mache ich falsch?
P: Du machst nichts falsch; du hast die Komponente gelöscht, aber das Programm hat sie noch nicht vom Bildschirm entfernt.
B: Wann wird sie entfernt?
P: Wenn du Control-J drückst, um das Bild neu zu zeichnen.
B: Ich versuch's...so, das wär's; aber die Komponente wurde nur zum Teil entfernt!
P: Entschuldige, ich habe vergessen: du musst jede Hälfte der Komponente einzeln löschen. Zeig' einfach nochmals auf CD.
B: Sehr gut...nun, was ist geschehen?
P: Du bist wieder im Analysis Mode: drück Control-Q.
B: Control...wo ist dieses Q? Ah hier...he, warum ist der Bildschirm plötzlich leer?
P: Du hast Q statt Control-Q eingegeben, so hast du das Programm verlassen. Es tut mir wirklich leid, aber wir haben alles verloren und müssen wieder von vorne beginnen.
B: [stöhnt] Können wir das auf nächste Woche verschieben?

(aus: W.M. Newman, R.F. Sproull: Principles of Interactive Computer Graphics, p. 443f.; Übers. des Verf.)

Solche schlechten Erfahrungen prägen die Haltung der Anwender gegenüber den benützten Systemen, aber auch gegenüber der EDV im allgemeinen. Die dadurch erzeugte oder verstärkte *negative Einstellung* erschwert es heute vielen potentiellen Anwendern, von den Vorteilen neuer Technologien profitieren zu können. Die Vermessung dürfte aber ein Berufsstand sein, dessen Zukunft vom Einsatz dieser Technologien abhängt [V+D 1987, Frank 1987]. Eine bessere Qualität der Interaktion zwischen Mensch und Computer sollte deshalb ein zentrales Anliegen von Ausbildung und Praxis im Vermessungswesen sein.

3. Wie entwickelt man «benützerfreundliche» Systeme?

«Software muss so entworfen werden, dass sie nicht das Arbeiten der Maschinen, sondern das Denken der Menschen optimiert» [Birnbaum 1985]

3.1. Entwurfsmethodik

Der Entwurf von Benützerschnittstellen ist «Arbeit an der Front», im Gegensatz etwa zum Entwurf von Algorithmen oder Datenstrukturen: Die Entscheide beeinflussen unmittelbar die Arbeitsweise der Benützer mit dem System. Was zählt ist schliesslich, wie die Benützer *tatsächlich* mit einem System zurechtkommen, nicht wie sich der Entwickler vorstellt, dass ein Idealbenützer sich verhalten wird.
Es ist oft schwierig, vorherzusagen, wie sich ein Entwurfsentscheid konkret auswirken wird. Bis heute gibt es *noch keine allgemeine Theorie,* die es erlauben würde, Voraussagen über die Eignung von Interaktions-Entwürfen zu machen. Interaktions-Gestaltung ist noch ebensosehr eine Kunst wie eine Wissenschaft. Immerhin liegen brauchbare theoretische Ansätze für Teilbereiche vor [Card et.al. 1980, Reisner 1981, Kieras und Polson 1985], und die Interaktions-Forschung ist auf dem Weg von einem beschreibenden Frühstadium zu einer fundierten Wissenschaft. Diese soll theoretische Modelle entwickeln, um Vorhersagen über die Eignung von Benützerschnittstellen machen und praktikable Entwurfsmethoden bereitstellen zu können.

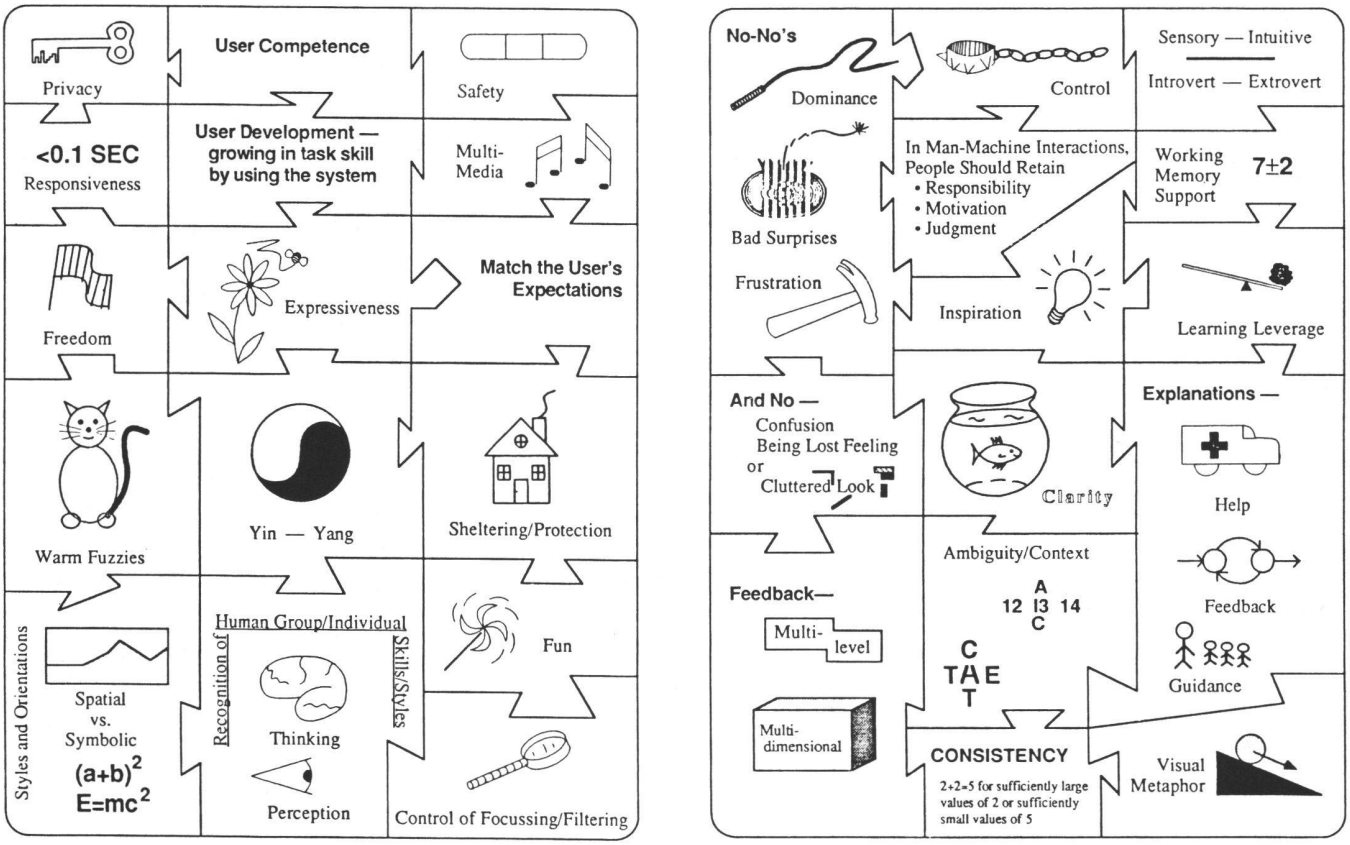

Abb. 1: «Anforderungen» an Benützerschnittstellen (aus ACM SIGCHI Bulletin, vol. 18 No. 2).

Eine Systematisierung der Entwurfsmethodik ist wichtig und kann nicht durch experimentelles Vorgehen oder durch den Antrieb neuer Technologien ersetzt werden. Denn Prototypen und Experimente lassen oft neue Lösungsmöglichkeiten ausseracht, und die technologische Entwicklung treibt uns zwar weiter, aber nicht unbedingt in die gewünschte Richtung.

3.2. Interaktions-Grundsätze

In den vergangenen 10 bis 15 Jahren sind von Informatikern und Psychologen verschiedene Sammlungen von Richtlinien und Grundsätzen für die Interaktions-Gestaltung verfasst worden. Sie stützen sich z.T. auf mehr oder weniger fundierte Faustregeln, wie etwa

- der Mensch kann nur 7±2 Dinge im Kurzzeit-Gedächtnis behalten;
- die Reaktionszeit eines Systems sollte unter 0,1 Sekunden liegen;
- Erkennen und Zeigen ist einfacher als Erinnern und Eintippen.

Als eher spielerischer Denkanstoss illustriert Abbildung 1 einige dieser vagen Anforderungen an Benützerschnittstellen.
Statt zu versuchen, «benützerfreundlich» oder andere Schlagworte zu definieren, fassen wir hier, ohne Anspruch auf Vollständigkeit, die wichtigsten Interaktions-Grundsätze aus einigen Quellen [Nievergelt 1982, Smith et.al. 1982, Shneiderman 1987] zusammen. Sie helfen einerseits dem Entwickler beim Systementwurf, andererseits aber auch dem Benützer bei der Evaluation von Systemen:

1. Es ist ein *Interaktions-Modell* zu wählen und dem Benützer explizit klarzumachen. Dies bedeutet, dass bewusst eine bestimmte Vorstellung festzulegen ist, die der Benützer vom System bekommen soll und mit der er sich das Systemverhalten erklären kann.
2. *Konsistenz:* Ähnliche Situationen sollen ähnliche Handlungen erfordern, damit das Systemverhalten vorhersehbar wird. Befehle, Meldungen und Dokumentation sollen die gleiche Terminologie verwenden.
3. Einfaches soll *einfach* sein, Kompliziertes *möglich.*
4. Die Schnittstelle soll den *Benützerwünschen angepasst* werden können: erfahrene Benützer sollen z.B. Abkürzungen verwenden können.
5. *Fehlerbehandlung:* Systeme sind nicht nur so zu entwerfen, dass Möglichkeiten für Eingabefehler eingeschränkt sind, sondern auch so, dass dem Benützer bei Fehlern geholfen wird.
6. Operationen sollen, wenn immer möglich, *umkehrbar* sein. Dies mindert die Angst der Benützer vor Fehlern und ermutigt sie, die Fähigkeiten eines Systems auszuprobieren.
7. Das *Kurzzeit-Gedächtnis* der Benützer soll nicht überlastet werden. Früher war es wichtig, beim Programmieren möglichst wenig Speicher und Prozessorzeit des *Computers* zu brauchen. Heute sind dies keine knappen Güter mehr, und Programme sollten deshalb bezüglich der Arbeitszeit des *Benützers* und der Belastung *seines* Kurz- und Langzeitgedächtnisses optimiert werden.
8. Der Benützer soll selbst die *Kontrolle ausüben* können. Wenn die Gewohnheit des Programmierers, dem Computer Befehle zu erteilen, auf den Benützer durchschlägt, so erzeugt dies Abwehrgefühle.
9. Der Benützer soll den *Systemzustand anschauen* können, ohne diesen dabei zu verändern. Das System soll dem Benützer jederzeit Antworten auf folgende Fragen anbieten:
   - Wo bin ich? (in welcher Umgebung, auf welche Daten wirken die Befehle)
   - Was kann ich hier tun? (welche Befehle sind möglich)
   - Wie kam ich hierhin? (welche Operationen brachten den Benützer hierhin)
   - Wohin kann ich gehen und wie? (welche Auswirkungen haben die angebotenen Befehle).

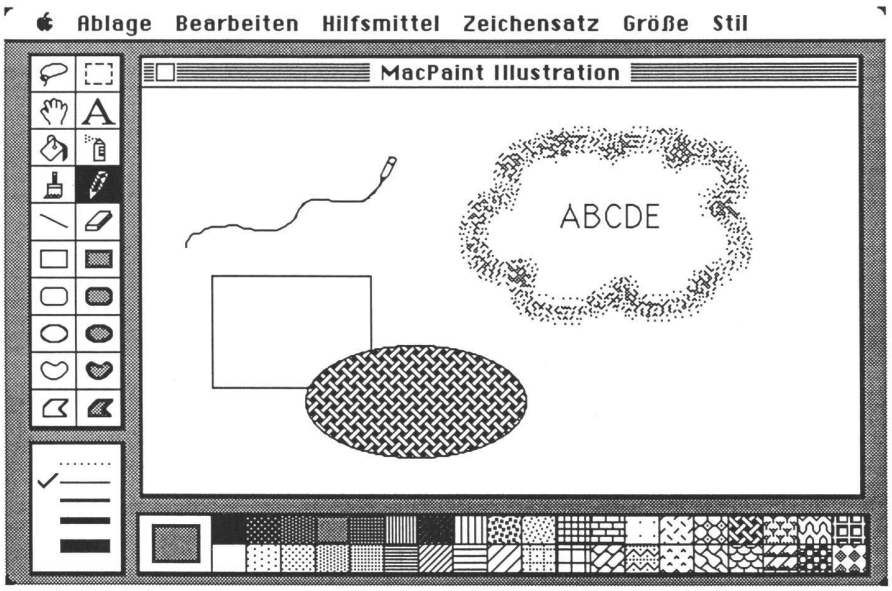

Abb. 2: Das Graphikprogramm MacPaint als Illustration der direkten Manipulation.

Alle diese Prinzipien müssen natürlich für jeden Systementwurf sinnvoll interpretiert und angepasst werden. Der gesunde Menschenverstand spielt immer eine zentrale Rolle beim Entwurf von Benützerschnittstellen. Dies wird wohl auch noch der Fall sein, wenn einmal umfassende formale und quantitative Entwurfs- und Beurteilungs-Methoden einsetzbar sein werden.

3.3. Interaktions-Stile und Interaktions-Mittel

Da sich Interaktivität aus der Programmierung heraus entwickelt hat, bieten interaktive Programme und Systeme oft *Befehlssprachen* an. Der Benützer gibt damit Operationsnamen an und bezeichnet durch *Parameter* die Objekte, auf die eine Operation wirken soll. Eine graphische Unterstützung fehlt oft oder dient nur zur Illustration der Ergebnisse. Eine Strukturierung der Befehle in Menüs entlastet zwar das Gedächtnis, bringt aber System und Benützer einander nicht näher. Der Benützer beschreibt *indirekt,* was geschehen soll; er schreibt dem Computer sozusagen einen Brief.

Im Gegensatz dazu können bei *direkter Manipulation* [Shneiderman 1983] Operationen direkt auf anschaulichen Darstellungen der Objekte ausgeführt werden (z.B. Kopieren und Löschen von Dateien, Zeichnen oder Skizzieren geometrischer Objekte usw., vgl. Abb. 2). Das Ergebnis jeder Operation wird unmittelbar sichtbar, und jeder Schritt kann rückgängig gemacht werden (UNDO-Befehl). So muss der Benützer keine Angst haben, einen Fehler zu begehen. Dies ist ein entscheidender Faktor beim Lernen der Systembedienung.

Direkte Manipulation wird hardware-seitig meist durch sogenannte Bitmap-Bildschirme und eine Maus unterstützt (Bsp.: Macintosh). Ein *Bitmap-Bildschirm* ist ein Anzeigegerät, dessen Bildebene in mehrere 10 000 Punkte («Pixel») unterteilt ist, die einzeln eingefärbt werden könnnen. Der jeweilige Zustand der Anzeige ist in einem speziellen Bereich des Hauptspeichers abgelegt; bei schwarz-weiss Bildschirmen wird je ein Bit des Speichers einem Pixel der Anzeige zugeordnet (daher «bitmap»). Objekte werden auf dem Bildschirm durch *«Icons»,* d.h. graphische Symbole dargestellt, und Befehle werden in Menüs angeboten, die nur dann erscheinen, wenn sie benötigt werden (sogenannte pop-up oder pull-down Menüs). Mit einer *Maus* oder einem anderen Zeigegerät können Icons angesprochen und bewegt werden. Ein Computer mit solchen Interaktions-Mitteln kostet heute soviel, wie ein alphanumerisches Terminal vor fünf Jahren.

4. Eine Anwendung: Geometrisches Konstruieren

4.1. Geometrisches Konstruieren in Landinformationssystemen

Ein Landinformationssystem (LIS) erhält seine Daten aus verschiedenen Quellen: Feldmessungen, Photogrammetrie, Fernerkundung, digitalisierte Pläne, andere Systeme usw. In einigen Fällen wird die *Datenerfassung* vollautomatisch erfolgen können, oft ist aber eine Interaktion mit dem Bearbeiter nötig. Besonders intensiv ist diese Interaktion dann, wenn Daten über *geometrische Objekte* eingegeben oder verändert werden, z.B. bei einer Parzellen-Mutation oder einer Absteckung. Solche Konstruktionsaufgaben wurden herkömmlicherweise graphisch auf Plänen ausgeführt. Wenn an die Stelle des Plans aber ein Informationssystem treten soll, müssen Konstruktionen direkt am Bildschirm ausgeführt werden können.

Von der Art der Gestaltung interaktiver Konstruktionsmethoden hängt, neben der Benützbarkeit des Systems, auch die *Konsistenz* seiner Informationen ab: Es soll z.B. nicht möglich sein, Parzellen zu konstruieren, deren Grenzlinien nicht geschlossen sind, oder sie so zu teilen, dass die Summe der Teilflächen nicht die alte Fläche ergibt, oder eine Strasse durch ein Haus zu legen. Das System muss also nicht nur Operationen zum Konstruieren anbieten, sondern auch dafür sorgen, dass durch diese Operationen das, in der Datenbank gespeicherte, Modell der Realität nicht inkonsistent wird. Auf diesen wichtigen Aspekt kann hier leider nicht weiter eingegangen werden (siehe dazu [Frank und Kuhn 1986]).

4.2. Interaktions-Modell: Konstruieren oder Entwerfen?

Beim Entwurf einer Benützerschnittstelle soll der Entwickler des Systems gemäss den in Abschnitt 3.2. zitierten Grundsätzen zuerst ein *Interaktions-Modell* festlegen. Vielfach wird ein solches in der gewohnten (manuellen) Arbeitsweise des Benützers gesucht und die Interaktion mit dem System wird dieser nachgebildet. So dient etwa beim Macintosh die Schreibtisch-Umgebung mit den gewohnten Handlungen des Öffnens, Kopierens und Wegwerfens von Dokumenten und Ordnern als Modell.

Im Fall des geometrischen Konstruierens liegt es nahe, die klassischen *«Zirkel und Lineal»-Konstruktionen,* mit denen der Ingenieur seit der Primarschule arbeitet, auch als Operationen am Bildschirm anzubieten: Definition einer Geraden durch zwei Punkte oder durch Punkt und Azimut, Kreis durch drei Punkte oder Punkt und Radius, Schnittkonstruktionen usw. In der Tat verwenden praktisch alle heute erhältlichen Systeme dieses Interaktions-Modell. Das Angebot an Grundkonstruktionen geht dabei mehr oder weniger weit über einen minimal nötigen Satz hinaus, so dass auch gewisse Flächenoperationen vorhanden sind, oder etwa die gemeinsamen Tangenten an zwei Kreise direkt konstruiert werden können. Abbildung 3 zeigt ein typisches Menü mit solchen Konstruktionsbefehlen.

Ist aber dieses Zirkel- und Lineal-Modell das einzig mögliche oder das beste Interaktions-Modell für geometrische Konstruktionen? Betrachten wir den manuellen Arbeitsablauf genau, so sehen wir, dass ein Konstrukteur fast immer zuerst eine *Skizze* erstellt, worin er das Ziel der Konstruktion grob darstellt und die *Bedingungen* der Aufgabenstellung (Abstände, Radien, rechte Winkel usw.) angibt. Dann muss er sich einen *Konstruktionsplan* zurechtlegen, um mit den zur Verfügung ste-

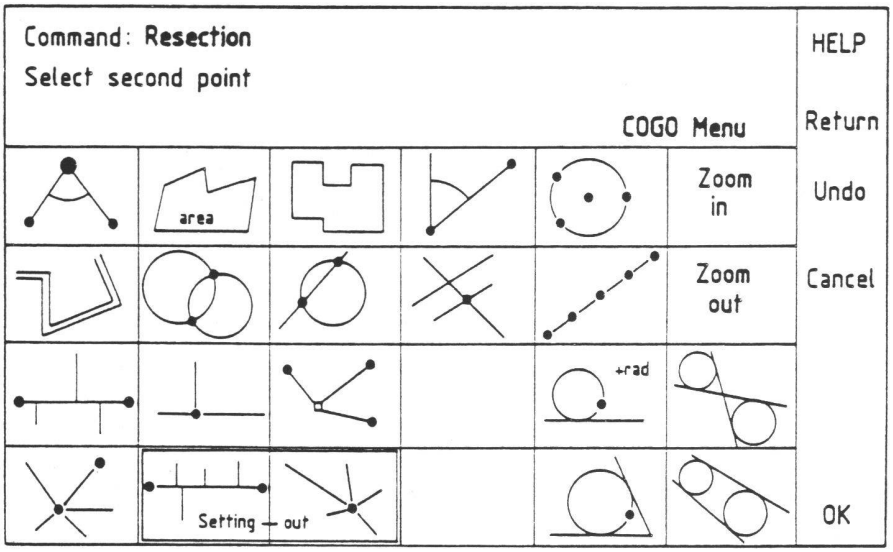

Abb. 3: Herkömmliche Menüfunktionen für geometrische Konstruktionen (aus [Wild 1985]).

henden Grundkonstruktionen schrittweise die gewünschte Situation konstruieren zu können.

Wenn dieses Vorgehen in der beschriebenen Weise automatisiert wird, so muss der Benützer die Lösung der Konstruktionsaufgabe weitgehend selbst suchen, das System hilft ihm fast nur beim Zeichnen. In vielen Fällen ist es zwar trivial, den Konstruktionsplan aufzustellen, manchmal kann aber eine Reihe von Konstruktionsschritten nötig sein, und das System bietet keinerlei Hilfen an, um diese zu planen. Insbesondere muss die Skizze, das zentrale Hilfsmittel bei der Planung, nebenher angefertigt werden. Unter- oder überbestimmte Probleme, die im Ingenieurwesen die Regel sind, können meist nicht behandelt werden. Ausserdem fehlen allgemeine Flächenoperationen, da solche eben auch mit Zirkel und Lineal nicht direkt ausführbar sind.

4.3. Ein Projekt am Institut für Geodäsie und Photogrammetrie (IGP) der ETHZ

Für das IGP-Projekt *LingLIS* (A *L*anguage with *I*nteractive *G*raphics for *LIS*) wurde nach einer eingehenden Problemanalyse ein anderes Interaktions-Modell gewählt [Kuhn 1986]: Der Benützer entwirft direkt am System eine Skizze der Aufgabenstellung, kann dazu Bedingungen über Abstände, Flächen usw. formulieren, und das System berechnet daraus eine Lösung. Das Umsetzen der Skizze in einen Konstruktionsplan entfällt, denn falls die Aufgabenstellung vollständig ist, gibt es keinen Grund, warum das System die Lösung nicht selbst finden könnte. Auf diese Art kann sich der Benützer darauf konzentrieren, *was* er konstruieren will und nicht *wie* das zu erreichen ist.

Das Macintosh-Programm *HILS* (*H*uman *I*nterface to *L*east *S*quares, [White 1988]) ist ein Prototyp für ein solches System. HILS erlaubt dem Benützer, im Stil der direkten Manipulation Punkte und Linien zu zeichnen und geometrische Bedingungen über diese Elemente anzugeben. Abbildung 4 zeigt eine, mit der Maus erzeugte, «Skizze» einer Mutation und das Menü mit den angebotenen Bedingungen. Je nach den in der Skizze ausgewählten Objekten können die dafür in Frage kommenden Bedingungen ausgewählt werden (z.B. «Area» für vier ausgewählte Punkte). In einem Dialog gibt der Benützer dann die Werte für eine Bedingung an. Ein Editor erlaubt es, Bedingungen nachträglich zu verändern. Bedingungen können auch direkt als Formeln eingetippt werden.

Neue Punkte und Linien und neue Bedingungen können in beliebiger Reihenfolge hinzugefügt werden. Wenn der Benützer den Befehl «Adjust» wählt, berechnet das Programm eine Lösung mittels einer Ausgleichung nach kleinsten Quadraten und stellt das Ergebnis dar. Eine numerische Unterbestimmung tritt nie auf, da das Programm die skizzierte Figur bereits als Näherungslösung betrachtet, die durch die angegebenen Bedingungen verbessert werden soll. Überbestimmungen werden durch die Ausgleichung behandelt. Der Einfluss der einzelnen Bedingungen kann dabei durch Gewichte oder Varianzen gesteuert werden.

Das Programm wird von seinem Autor, M. White, im Auftrag des IGP weiterentwikkelt und dabei noch stärker auf die Bedürfnisse der Vermessung ausgerichtet. Die theoretischen Untersuchungen über die Interaktion beim geometrischen Konstruieren werden gegenwärtig abgeschlossen. Sie ergaben, neben dem besprochenen Interaktions-Modell, eine formale Sprache für die vollständige Beschreibung von Konstruktionsaufgaben. Darauf aufbauend wurde ein allgemeiner Geometrie-Editor entworfen, der z.B. in einem Landinformationssystem eingesetzt werden kann.

5. Schluss

Es ging in diesem Beitrag nicht darum, das Thema Interaktion systematisch zu behandeln. Vielmehr sollten folgende Fragen gestellt und einige stichwortartige Antworten darauf versucht werden:

- Wo liegen die Probleme bei der Interaktion zwischen Benützer und System?
- Wie sehen heutige Lösungsansätze aus?
- Welche Arbeiten wurden auf diesem Gebiet unter Professor Conzett am IGP unternommen?

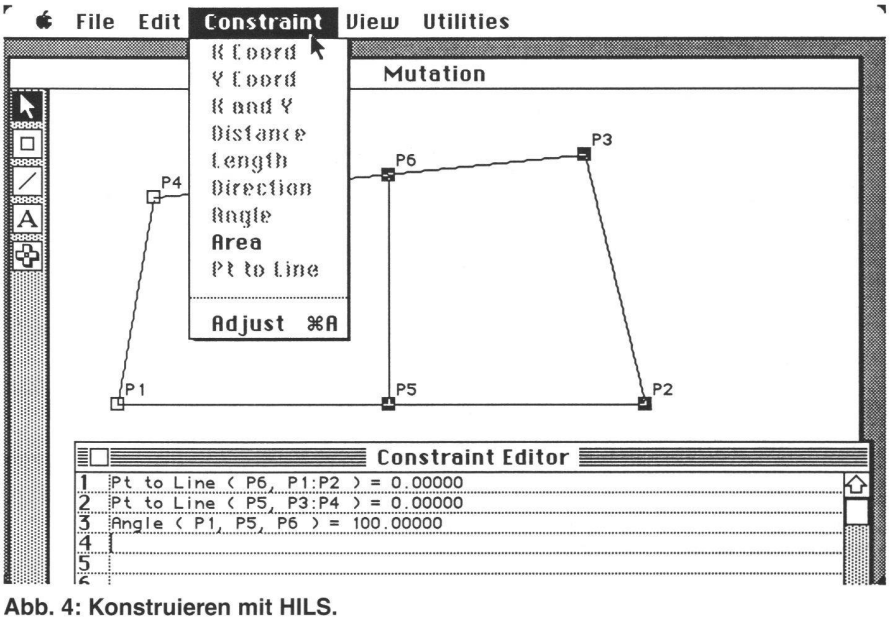

Abb. 4: Konstruieren mit HILS.

Partie rédactionnelle

Die beabsichtigten Hauptaussagen waren, dass

- Interaktion wirtschaftlich, psychologisch und sozial bedeutsam ist
- bessere Benützerschnittstellen mit heutigen Mitteln möglich sind
- Interaktions-Entwürfe gründliche Problemanalysen verlangen.

Literatur:

Bernet, J. 1986: Planen mit CAD – Voraussetzungen und Auswirkungen; Schweizer Ingenieur und Architekt; Nr. 23, 1986; pp. 575–586.

Birnbaum, J.S. 1985: Toward the Domestication of Microelectronics; ACM Communications; vol. 28, No. 11, November 1985; p. 1228.

Brügger, B. 1988: Softwarekonzepte für die Datenerfassung im Feld; VPK 11/88 (in diesem Heft).

Card, S.K., Moran, T.P., Newell, A. 1980: The Keystroke-Level Model for User Performance Time with Interactive Systems; ACM Communications; vol. 23, No. 7, Juli 1980; pp. 396–410.

Conzett, R. 1987: Automatische Datenverarbeitung in der Vermessung; Vorlesungs-Skript, ETH Zürich 1987.

Frank, A.U. 1987: Geo-Information Engineers: Surveyors in the Information Age; FIG XVIII. Congress Toronto, Bericht 102.1.

Frank, A.U., Kuhn, W. 1986: Cell Graphs – A Provable Correct Method for the Storage of Geometry; ETH Zürich, IGP Bericht Nr. 119.

Kieras, D.E., Polson, P.G. 1985: An approach to the formal analysis of user complexity; Int. Journal of Man-Machine Studies; vol. 22; pp. 365–394.

Kuhn, W. 1985: Zur Entwicklung Interaktiver Programme und Systeme; VPK 2/85; pp. 44–49.

Kuhn, W. 1986: LIS and the User – Looking for Easier Ways to do Geometric Constructions; FIG XVIII. Congress Toronto, Bericht P 305.2.

Nickerson, R.S. 1981: Why interactive computer systems are sometimes not used by people who might benefit from them; Int. Journal of Man-Machine Studies; vol. 15, 1981; pp. 469–483.

Nievergelt, J. 1982: Errors in Dialog Design and how to avoid them; in: Nievergelt, J. et.al. (Eds.), Document Preparation Systems; North Holland 1982.

Reisner, P. 1981: Formal Grammar and Human Factors Design of an Interactive Graphics System; IEEE Transactions on Software Engineering; vol. SE-7, No. 2, März 1981; pp. 229–240.

Shneiderman, B. 1983: Direct Manipulation: A Step Beyond Programming Languages; IEEE Computer; vol. 16, No. 8, August 1983; pp. 57–68.

Shneiderman, B. 1987: Designing the User Interface – Strategies for Effective Human-Computer Interaction; Addison-Wesley 1987.

Smith, D.C. et.al. 1982: Designing the STAR User Interface. BYTE April 1982; pp. 242–282.

V+D 1987: Die Zukunft unseres Bodens; Reform der Amtlichen Vermessung; Politischer Bericht.

White, R.M. 1988: HILS – Human Interface to Least Squares, Benützeranleitung; ETH Zürich, IGP Bericht Nr. 152.

Wigger, U. 1988: DATAUF – ein Programmsystem zur Aufbereitung und Verwaltung von Vermessungsdaten; VPK 8/88; pp. 427–433.

Wild 1985: System 9 – Preliminary Description; Wild Heerbrugg, Oktober 1985.

Adresse des Verfassers:
Werner Kuhn
Institut für Geodäsie und Photogrammetrie
ETH-Hönggerberg
CH-8093 Zürich

Softwarekonzepte für die Datenerfassung im Feld

B. Brügger

Im vierten und letzten Beitrag aus der Informationstagung vom 23./24. Oktober 1987 über die Ausbildung des Kulturingenieurs in Informatik im Vermessungswesen an der ETH Zürich geht es um die Datenerfassung im Feld. Schon seit einigen Jahren werden auf diesem Gebiet elektronische Rechner eingesetzt, ohne dass, wie in anderen Anwendungsgebieten, revolutionäre Arbeitserleichterungen und Steigerungen der Produktivität beobachtet wurden. In diesem Aufsatz werden Denkweisen des modernen Software Engineering angewandt, um nach Gründen dafür zu suchen.

In einem ersten Teil werden die Problemstellung der Datenerfassung für ein Landinformations-System (LIS) analysiert und darauf basierend ein Anforderungskatalog für die Erfassungssoftware aufgestellt. In einem zweiten Teil werden drei mögliche Modellierungen für Erfassungssoftware beschrieben und diskutiert. Das Vermessungsfachwissen, welches in ein solches Modell fliesst, entscheidet über die «Intelligenz» der Software und somit über ihren Gebrauchswert. Für jedes der drei Modelle wird beschrieben, wie dieses Fachwissen formuliert werden kann, wie die Art der Formulierung die «Intelligenz» des Modells beschränkt und wie sich der «Intelligenz-Grad» eines Modells auf die Erfassungsarbeit auswirkt.

1. Einleitung

Mit dem Aufkommen von elektronischen Theodoliten und Tachymetern wurde der Einsatz von Computern im Feld aktuell. Die primäre Aufgabe der Feld-Software ist die Verwaltung der anfallenden Daten. Ausserdem kann die verfügbare Rechenleistung zur Unterstützung der Feldarbeit eingesetzt werden. Die computerunterstützte Datenerfassung verspricht eine Rationalisierung der Arbeit und eröffnet neue Möglichkeiten im Feld.

Dieser Aufsatz zeigt, wie verschiedene Konzepte der Feld-Software sich auf die mögliche Unterstützung der Datenerfassung im Feld auswirken und wie sich die klassischen Arbeitsabläufe dabei ändern.

Im zweiten Abschnitt wird die *Problemstellung* bei der Datenerfassung näher be-

Institut für Geodäsie und Photogrammetrie
ETH-Hönggerberg, CH-8093 Zürich

Separata Nr. 140.