

Zeitschrift:	Vermessung, Photogrammetrie, Kulturtechnik : VPK = Mensuration, photogrammétrie, génie rural
Herausgeber:	Schweizerischer Verein für Vermessung und Kulturtechnik (SVVK) = Société suisse des mensurations et améliorations foncières (SSMAF)
Band:	85 (1987)
Heft:	3
Artikel:	Die ungleichen Gesprächspartner : Zur Benutzerfreundlichkeit von Dialog-Software
Autor:	Kistler, Valérie
DOI:	https://doi.org/10.5169/seals-233438

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

Download PDF: 25.01.2026

ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>

Die ungleichen Gesprächspartner

Zur Benutzerfreundlichkeit von Dialog-Software

Valérie Kistler

Selbst unter EDV-Spezialisten herrschte bisher Unklarheit darüber, was mit Benutzerfreundlichkeit im Bereich von Software genau gemeint ist. Vielfach diente der Begriff auch bloss als werbewirksames Verkaufsargument.

In einem umfangreichen Projekt erarbeitete der Lehrstuhl für Arbeits- und Organisationspsychologie an der ETH Zürich einen Kriterienkatalog für die Gestaltung von benutzerfreundlichen Mensch-Maschine-Dialogen. Es ist ein Konzept, das mit dem traditionellen Verständnis von Benutzerfreundlichkeit aufräumt.

Jusqu'à ce jour et même dans les rangs des spécialistes en traitement électronique des données, les désaccords étaient grands concernant la définition de la convivialité dans le domaine de la programmation. Bien souvent, ce terme n'était utilisé qu'à des fins de publicité.

Dans le cadre d'un important projet, la chaire de psychologie du travail et de l'organisation de l'EPF de Zurich a développé un catalogue de critères permettant de créer des dialogues homme-machine conviviaux. Sa conception de convivialité rompt radicalement avec le sens traditionnel donné à ce terme.

Bei der Hardware hat es ein Benutzer noch relativ leicht: Dank genauen Ergonomie-Richtlinien kann er schnell man selbst überprüfen, wie es um die Benutzerfreundlichkeit seines Computers steht. Schwieriger wird es hingegen bei der Software: Weil hier die Software-Ergonomie noch zuwenig objektive Kriterien für die Gestaltung des Mensch-Maschine-Dialoges zur Hand hat, ist die Beurteilung von Benutzerfreundlichkeit bisher notgedrungen dem Ermessen der Software-Hersteller überlassen worden – obwohl auch sie oftmals nicht wissen, was unter Benutzerfreundlichkeit genau zu verstehen ist.

«Es hat sich gezeigt, dass in der Praxis unter benutzerfreundlichen Dialogsystemen häufig Systeme verstanden werden, die mit Begriffen wie „narrensicher“, „leicht und schnell erlernbar“ oder „einfach zu bedienen“ charakterisiert werden können», sagt Philipp Spinas, Mitarbeiter am Lehrstuhl für Arbeits- und Organisationspsychologie (LAO) der ETH Zürich.

Analysen des LAO haben ergeben, dass die «Benutzerfreundlichkeit» derartiger Systeme weitgehend auf dem Versuch beruht, ihre Komplexität zu verringern. Dies kann bei der Dialoggestaltung auf zwei Arten erreicht werden:

– indem man die Anwendungsmöglichkeiten des Systems stark reduziert (womit der Computer nur noch als Ein-

zweck-Werkzeug für wenige, isolierte Teilaufgaben eines Arbeitsablaufes zu gebrauchen ist) und/oder
– indem man dem Benutzer, in seinem Handlungsspielraum eingeschränkt, durch eine starr vorgegebene Abfolge von Dialogschritten kaum Möglichkeiten lässt, aktiv in den Dialogablauf einzutreten.

Gemeinsam ist derartigen Dialogsystemen eine Grundstruktur, die aus verschiedenen, hierarchisch aufgebauten Menüs besteht, gefolgt von einer Anzahl Eingabemasken oder Informationsbildern.

Die Software-Entwickler bauen solche «Kunstgriffe» vielfach mit der durchaus gutgemeinten Absicht in ihre Programme ein, auf diese Weise den EDV-Anfänger nicht zu überfordern. Und tatsächlich schaffen sie damit einfache, schnell erlernbare Dialoge, als deren Vorteil der geringe Schulungsaufwand gepriesen wird. Nur: Ein derartig konzipiertes Dialogsystem wird von Benutzern, die intensiv damit arbeiten und deren EDV-Kenntnisse durch die praktische Anwendung schnell wachsen, bald einmal als Hemmschuh empfunden. «Die deutlich erkennbare Führung des Systems wird bloss in den ersten Wochen geschätzt. Sobald der Benutzer etwas geübt ist, fühlt er sich in einem systemgesteuerten Dialog nur noch als Bediener, der auf Signale zu reagieren hat», sagt Spinas.

Anders ausgedrückt: In den letzten Jahren sind häufig Systeme entwickelt worden, die für den sporadischen Gebrauch und für Anfänger gedacht waren, in Wirklichkeit

aber von Fortgeschrittenen täglich mehrere Stunden benutzt wurden.

Gefragt sind Dialoge für Profis

Dass geübte Benutzer nicht einfache, systemgesteuerte Dialoge wünschen, sondern komplexere, dafür mehr Freiheitsgrade gewährende Dialogsysteme vorziehen, konnte eine in zwei Betrieben durchgeführte Pilotstudie des LAO deutlich belegen: Danach gaben 84 Prozent der befragten Benutzer eindeutig der komplexeren Dialogvariante den Vorzug – selbst wenn damit ein grösserer Schulungsaufwand verbunden war. Spinas: «Dieses Ergebnis kann als deutliche Absage an eine Auffassung von Benutzerfreundlichkeit verstanden werden, die narrensichere, systemgesteuerte Dialoge beinhaltet».

Das Urteil der Benutzer legt also ein ganz anderes Verständnis von Benutzerfreundlichkeit an den Tag. Aus Benutzersicht kann ein Dialogsystem dann als benutzerfreundlich bezeichnet werden, «wenn bei einer angemessenen Anzahl von Freiheitsgraden für unterschiedliche Vorgehensweisen der Handhabungsaufwand des Computers in einem als zufriedenstellend erlebten Verhältnis zu seinen Anwendungsmöglichkeiten – und dem damit verbundenen Nutzen – steht», heisst es in einer Zusammenfassung der LAO-Untersuchungsergebnisse.

Wie muss demzufolge Dialogsoftware aufgebaut sein, damit sie einem so definierten Verständnis von Benutzerfreundlichkeit entspricht, damit sie den Forderungen der Benutzer nach vielfältigen Anwendungsmöglichkeiten und vermehrten Freiheitsgraden gerecht wird?

Mitarbeiter des LAO haben nach einer Reihe von Studien und Untersuchungen, die sich über einen Zeitraum von zweieinhalb Jahren erstreckten, sieben Regeln für die Gestaltung von Dialogschnittstellen entwickelt. Sie bauen auf allgemein- und arbeitspsychologischen Erkenntnissen auf – wohlwissend, dass es zwar den Benutzer eines Computersystems nicht gibt, dass aber gewisse Merkmale der Informationsaufnahme und -verarbeitung allen Menschen gemeinsam sind.

Sieben Kriterien für Software-Designer

Für eine benutzerfreundliche Dialoggestaltung von System- und Anwendungsprogrammen hat die LAO-Forschungsgruppe die sieben folgenden Kriterien erarbeitet:

Transparenz:

– Die Struktur des Dialogsystems sollte prägnant und durchschaubar sein, damit sich der Benutzer ein inneres Modell davon aufbauen kann. Die Zergliede-

Partie rédactionnelle

rung der Struktur in kleine Teilschritte ist zu vermeiden.

- Der Benutzer sollte erkennen können, ob ein eingegebener Befehl behandelt wird oder ob das System auf weitere Eingaben wartet.
- Bei längeren Vorgängen solten Zwischenstandsmeldungen erscheinen.

Konsistenz:

- Das Systemverhalten muss für den Benutzer vorhersehbar und einschätzbar sein, auch in bezug auf Antwortzeit und Ausgabeform. Erwartungen sollten erfüllt, Überraschungseffekte vermieden werden. Ähnliche Aktionen sollten somit auch ähnliche Ausführungen bewirken. Diese Maxime verfolgt beispielsweise das Schweizer Unternehmen Delta Software Technologie, welches in der Entwicklung von Werkzeugen für Applikations-Software als Nummer Eins in Europa gilt, bereits seit Jahren. «Prinzip der geringsten Verwunderung» nennt Delta-Direktor Reinhold Thurner den Firmen-Leitsatz für die Entwicklung ihrer Tools.

Toleranz:

- Das System verzichtet auf präzise Eingabe-Erfordernisse.
- Operationen sollten ohne Verlust von Daten reversibel sein.
- Der Benutzer sollte die Möglichkeit haben, an bestimmten Punkten, die entweder vom System fix gesetzt oder vom Benutzer gewählt sein können, den Dialoglauf zu unterbrechen. Eingaben zu ändern und danach den Dialog wieder fortzusetzen.

Unterstützung:

- Sowohl für inhaltliche als auch den Dialoglauf betreffende Fragen des Benutzers sollte das System jederzeit abrufbare Hilfen (z.B. Help-Taste) zur Verfügung stellen. Diese sollten gegenüber anderen Befehlen bevorzugt behandelt werden.
- Das System sollte Rückfragen des Benutzers erlauben und darauf ausführliche Antworten geben.

Kompatibilität:

- Diese Richtlinie stützt sich auf die Erkenntnis, dass das Kurzzeitgedächtnis des Menschen eine sehr begrenzte Kapazität hat: Es können nur vier bis sieben Informationen gleichzeitig und auch nur für eine Dauer von etwa zehn Sekunden behalten werden. Darum ist bei der Darstellungsform für Einzelinformationen ebenso wie für ganze Bilder darauf zu achten, dass diese mit entsprechenden gedruckten Vorlagen übereinstimmen.
- Die im System verwendete Sprache und die Begriffe sollten nicht aus EDV-Kürzeln bestehen, sondern dem Sprachgebrauch und dem Fachgebiet des Benutzers entnommen sein.

Flexibilität und Individualisierbarkeit

- Dieses Kriterium bezieht sich auf die bereits oben genannten LAO-Untersuchungen. Die Steuerung des Dialogablaufes sollte so weit wie möglich dem Benutzer überlassen werden. Dadurch hat er die Möglichkeit, sein Vorgehen individuell zu gestalten und Freiheitsgrade zu nutzen.
- Der Benutzer sollte das Tempo der Abfolge der einzelnen Dialogschritte selber bestimmen können.

Partizipation:

- Damit wird der frühzeitige Einbezug des Benutzers bei der Entwicklung und der Einführung von Dialogsystemen gefordert, um optimale, auf seine Bedürfnisse und Ansprüche abgestimmte Lösungen zu erreichen.

Benutzerbezogene Software-Entwicklung

Zum Kriterium Partizipation sind von der LAO-Forschungsgruppe in zwei schweizerischen Betrieben die Tätigkeiten von Sachbearbeitern untersucht worden: Neben der Beobachtung der Arbeitsabläufe und der Analyse der vorgefundenen Dialogstrukturen wurden die Angestellten in einem halbstandardisierten Interview nach ihrer Einschätzung des verwendeten Dialogsystems befragt.

Beim ersten Betrieb handelte es sich um eine Versicherungsgesellschaft, in der 21 Sachbearbeiter, welche Versicherungs-Offerten zu erstellen und Korrespondenz zu erledigen hatten, befragt wurden. Die Sachbearbeiter waren seinerseits bei der Entwicklung des neuen Computer-Arbeitssystems kaum miteinbezogen worden, sechs Monate vor deren Einführung fand lediglich eine Orientierung statt. Die Ausbildung hatte sich auf die Bedienung des Systems beschränkt, um – wie es im späteren LAO-Bericht heißt – «reibungsloses Abarbeiten vorgegebener Abläufe sicherzustellen.»

Anders in Betrieb 2: In diesem Handelsunternehmen bestand die Aufgabe der 25 Sachbearbeiter in der Erfassung und Bearbeitung von Aufträgen und der Erledigung von Korrespondenz. Hier wurden die zukünftigen Benutzer zwölf Monate vor der geplanten Einführung des Systems umfassend orientiert. In der Entwicklungsphase konnten sie Anregungen und Wünsche vorbringen. «Unser System umfasst heute klar definierte Lösungen, die praktisch alle auf Vorschläge unserer Mitarbeiter zurückzuführen sind», sagt Peter Oeschger, Geschäftsführer dieses Handelsunternehmens.

In der nachfolgenden Befragung der Sachbearbeiter beider Betriebe zeigte sich, dass die Angestellten der Handelsfirma ihr neues Arbeitssystem eindeutig positiver

beurteilten als die Sachbearbeiter von Betrieb 1, die kaum in die Entwicklung und Einführung des Bildschirmsystems einzogen worden waren.

Auf die grosse Bedeutung einer partizipativen Entwicklung von Dialog-Software weist jedoch nicht nur die Arbeitswissenschaft hin – das Bewusstsein und der Wille, den zukünftigen Benutzer in den Gestaltungsprozess von Software miteinzubeziehen, sind mittlerweile auch bei den Software-Entwicklern immer spürbarer vorhanden: Es gibt Beispiele von Unternehmen, deren Programmierer zuerst monatelang als Sachbearbeiter arbeiten, um deren Bedürfnisse genauer kennenzulernen und in die spätere Dialoggestaltung einbringen zu können.

Eine derart benutzerbezogene Dialogentwicklung ist jedoch ein nie endender Prozess. Durch die Benutzung erwachsen neue Anforderungen, die nachträglich ins Programm eingebaut werden müssen. «Es wäre darum falsch, fixfertige Programme entwickeln zu wollen», sagt Reinhold Thurner von Delta Software Technologie. «Als bewährtes Vorgehen gilt, einen gut änderbaren ersten Release zu erstellen und die damit gesammelten Erfahrungen in einen zweiten und dritten Release einzubauen». Was heißt: Entwicklung für die Weiterentwicklung der Entwicklung . . .

Subjektive Objektivität

Die sieben vom LAO aufgestellten Kriterien für die Gestaltung von benutzerfreundlichen Mensch-Maschine-Dialogen dürfen nach Ansicht von LAO-Mitarbeiter Philipp Spinas nicht nur auf die Software allein bezogen werden: «Sie müssen immer auch im Kontext gesehen werden zum jeweiligen Benutzer und seinen Qualifikationen, im Kontext zur Aufgabe, welche mit Hilfe des Computers erledigt werden soll und zur Häufigkeit, mit der die Software benutzt wird».

Auf alle diese Einflussgrößen bezogen kann Benutzerfreundlichkeit in mehr oder weniger grossem Massen erreicht werden. Die Spannweite der Realisierung von Benutzerfreundlichkeit ist denn auch entsprechend gross: Sie reicht von der Anpassung der Software an allgemeine körperliche und geistige Eigenschaften des Menschen bis zur Berücksichtigung von individuellen Unterschieden ihrer Benutzer.

Die Beurteilung der Benutzerfreundlichkeit einer Software umfasst somit immer die Beurteilung einer ganzen Fülle von Eigenschaften. Die Gesamtheit dieser Eigenschaften lässt sich jedoch nur bis zu einem gewissen Grade technisch überprüfen. Je nach den Bedürfnissen, Ansprüchen und Erwartungen, die ein Benutzer an die Dialogsoftware stellt, bleibt in der Beurteilung immer auch eine Spur Subjektivität miteingeschlossen. «Gerade die

Software-Ergonomie bemüht sich jedoch darum, solche als subjektiv empfundenen Eigenschaften von Benutzerfreundlichkeit durch Experimente und wissenschaftliche Untersuchungen in Empfehlungen zur Gestaltung von Software umzusetzen, die dann nicht nur objektiv als richtig gelten, sondern auch im subjektiven Urteil ihrer Benutzer als richtig und positiv angesehen werden», sagt Joachim Grieser, Direktor des Institutes für Wirtschaftsinformatik an der Universität Bern. «Im Vergleich zur Hardware-Ergonomie, wo mittlerweile recht gute, wissenschaftlich fundierte Gestaltungsempfehlungen bestehen, stehen wir in der Software-Ergonomie allerdings noch ganz am Anfang.»

Literaturverzeichnis

- Spinas/Troy/Ulrich: Leitfaden zur Einführung und Gestaltung von Arbeit mit Bildschirmsystemen. CW Publikationen, München, 1983.
- E. Ulrich: Aspekte der Benutzerfreundlichkeit. Erschienen in: Remmeli/Sommer (Hrsg.): Arbeitsplätze morgen. Teubner, Stuttgart, 1986.
- P. Spinas: Benutzerfreundlichkeit und die Analyse von Dialog-Strukturen. Erschienen in: Spectrum Nr. 14. Österreichische Gesellschaft für betriebliche Ausbildung (ÖGA), Wien, 1985.
- E. Ulich: Computerunterstützte Büroarbeit – Konsequenzen für die Personalpolitik. Ebenfalls erschienen in Spectrum Nr. 14.

Software-Test unter Beschuss

In der Bundesrepublik Deutschland ist die Diskussion um verbindliche Normen für die Gestaltung des Mensch-Maschine-Dialoges in vollem Gange: Seit Ende 1984 liegt beim Deutschen Institut für Normung (DIN) in Berlin unter der Nummer 66 234 ein Entwurf, der Richtlinien für die Gestaltung von Dialogsystemen definiert. Danach sollen Dialogsysteme den folgenden Grundsätzen entsprechen: Aufgabenangemessenheit, Selbsterklärungsfähigkeit, Steuerbarkeit, Erwartungskonformität und Fehlerrobustheit.

Dieser Norm-Entwurf ist gegenwärtig der Öffentlichkeit zur Prüfung und Stellungnahme übergeben. Es sei hingegen nicht absehbar, dass in der nächsten Zeit etwas Endgültiges in dieser Sache beschlossen werde, heisst es am Berliner Normen-Institut.

In der Zwischenzeit haben sich in der Bundesrepublik verschiedene Hersteller, Anbieter und Anwender von Software – darunter auch Firmen wie IBM, DEC und Siemens – zur Gütegemeinschaft Software zusammengeschlossen und eigene Qualitätsbestimmungen für Anwenderprogramme aufgestellt. Damit wollen sie den Benutzern einen Massstab in die Hand geben, um «gute» Software von «schlechter» unterscheiden zu können.

An autorisierten Prüfstellen der Gütege-

meinschaft Software wird seit Oktober letzten Jahres Anwendungssoftware nach DIN V 66 282 aufgrund der Produktebeschreibung des Herstellers geprüft. Diese wiederum muss den Anforderungen von DIN V 66 285 entsprechen. Untersucht werden die Programme im wesentlichen auf die folgenden Punkte: Vollständigkeit der Funktionsbeschreibungen, Korrektheit, benutzerbezogene Robustheit, Einheitlichkeit und Verständlichkeit der Fehlermeldungen, Einheitlichkeit des Dialogverhaltens sowie Konsistenz.

Programme, die den Test bestehen, erhalten vom Güteausschuss eine Verleihungsurkunde und dürfen fortan das DIN-geprüft-Zeichen sowie das RAL-Gütezeichen Software tragen. 15 Programme seien seit Bestehen dieses Software-Testverfahrens bisher mit der Qualitätsplakette versehen worden, meldete unlängst die deutsche Informatik-Zeitung «Computerwoche».

Diese Art von Software-Prüfung ist jedoch schon kurz nach ihrer Einführung zunehmend unter Beschuss geraten. Bereits im Januar 1986 äusserte der Zentralverband des Deutschen Handwerks ernste Bedenken gegen das «Qualitätssiegel». Ein halbes Jahr später lief auch die Vereinigung Deutscher Softwarehersteller (VDS) gegen das RAL-Prüfverfahren Sturm: Das Gütezeichen Software sei nichts anderes als der geschickte Versuch der prüfenden Firmen, einen neuen Markt aufzutun, für den kein Bedürfnis bestehe, liess VDS-Geschäftsführer Roland Blickmann in der «Computerwoche» verlauten. Zudem werde der Schein erzeugt, als ob nur noch «gütegezielte Software» auch gute Software sei. Damit werde den kleinen Anbietern das Wasser abgegraben.

Gewettet wurde von Seiten des VDS auch gegen die «zu laschen» Prüfkriterien und die zu hohen Test-Kosten: Die «Computerwoche» nennt Testgebühren «zwischen 10 000 und 60 000 Mark» je nach Umfang des zu prüfenden Programms. Auch in der Schweiz herrscht gegenüber eines solchen Software-Prüfverfahrens Skepsis. Die Gesellschaft Schweizerischer EDV-Dienstleistungsunternehmen und Software-Hersteller (GES) jedenfalls würde eine ähnliche Teststelle für die Schweiz nicht begrüssen. Willi Vogt, verantwortlich für den Fachbereich Software-Produkte innerhalb der GES, begründet: «Die Zeit ist noch lange nicht reif, um bereits verbindliche Kriterien für die Software-Entwicklung aufzustellen.»

Erwünscht wären von Seiten der GES jedoch Empfehlungen für Software-Standards. «Damit könnte man einerseits die industrielle Herstellung von Software-Produkten erleichtern und ihre Verbreitung fördern. Andererseits müsste auch weniger Zeit in die Einführung von Mitarbeitern investiert werden», sagt GES-Präsident Jürg B. Winter.

Der über 50 Firmenmitglieder zählende Branchenverband GES würde im Bereich der Benutzeroberfläche eine einheitlichere Gestaltung der Dialogstruktur, der Dialogabläufe sowie der Funktionstasten begrüssen. Im Bereich der Schnittstellen wäre eine Standardisierung der Datenformate wünschbar. «Solche Empfehlungen müssten jedoch nicht nur wie bisher in den einzelnen Kantonen diskutiert, sondern auch auf nationaler Ebene angegangen werden», meint GES-Präsident Winter.

Das System XS-2: Modell eines Software-Standards

Das integrierte, interaktive Dialogsystem XS-2 stellt den Versuch dar, den Mensch-Computer-Dialog auf systematische Art zu standardisieren und allen Klassen von Benutzern eine einheitliche Schnittstelle zur Verfügung zu stellen. Dadurch soll XS-2 auf viele verschiedene Applikationsprogramme anwendbar sein.

Das System XS-2 ist an der ETH Zürich – aufbauend auf den von ETH-Professor Jürg Nievergelt entwickelten Vorläufer-Systemen XS-0 und XS-1 – von Dr. Jan Stelovsky, Oberassistent am Institut für Informatik der ETH Zürich, ausgearbeitet worden. Seit 1983 wird es an der Zürcher ETH sowie im Forschungslabor der BBC Baden experimentell verwendet.

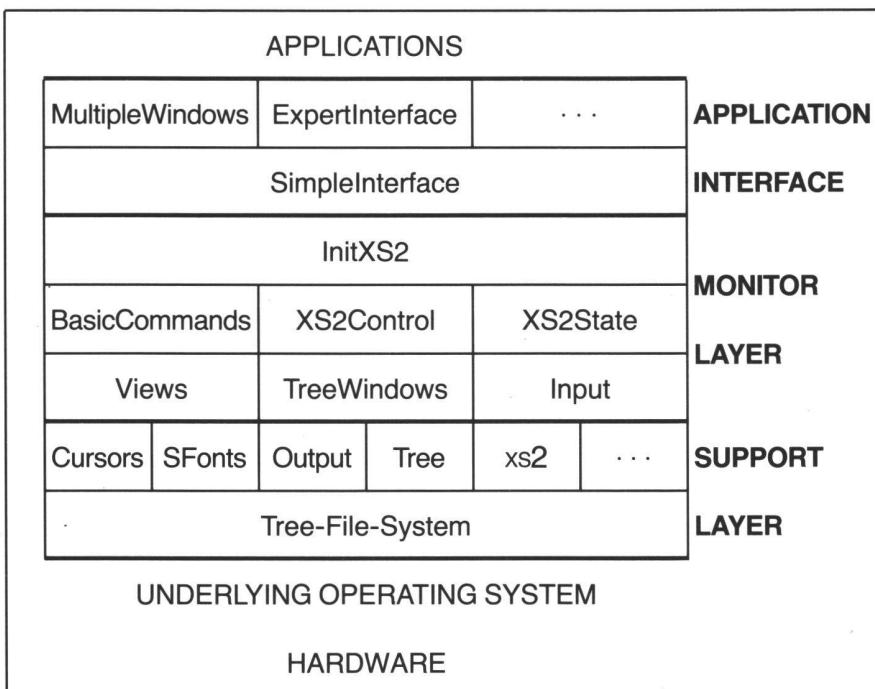
Das System XS-2 ist nach den folgenden Prinzipien aufgebaut:

- Durchschaubarkeit des Systemzustands
- Einfachheit und Einheitlichkeit der verwendeten Konzepte
- Effizienz
- Anpassbarkeit auf die individuellen Bedürfnisse der Benutzer
- Ausbaubarkeit des Systems.

Die Benutzerfreundlichkeit des Systems wird durch die Realisierung dieser Prinzipien erreicht: In XS-2 ist der Systemzustand in Daten, Befehle und Dialoggeschichte aufgeteilt. Diese Komponenten des Systems sind hierarchisch strukturiert und in separaten Bildschirmfenstern immer sichtbar. Mit einer kleinen Anzahl von universellen Befehlen, die immer aktiv sind, können Bewegungen und Sichtänderungen an diesen Strukturen ausgeführt werden. Dadurch kann sich der Benutzer jederzeit über den Systemzustand informieren, ohne ihn dabei zu verändern.

Die Befehlssprache des Systems XS-2 wird in Form von immer gleich strukturierten Befehlsbäumen dargestellt, die Befehle samt allen ihren Parametern enthalten. Es braucht somit nicht zwischen Befehlen und Parametern unterschieden zu werden. Eine geringe Anzahl von Befehls-

Partie rédactionnelle



Die Software-Struktur des interaktiven Systems XS-2.

typen klassifiziert die Befehle. «Daher sprechen alle Anwendungsprogramme sowie der Kern des Systems die gleiche Sprache», sagt Jan Stelovsky. Die ganze Syntax der Befehlssprache ist im Befehlsfenster sichtbar und kann zur Eingabe verwendet werden. Der Benutzer kann die einzelnen, im Kommandofenster sichtbaren Befehle mit der Maus auswählen. Er kann aber auch von einer beliebigen

Stelle in XS-2 aus jeden Befehl ausführen, dessen Name oder dessen Abkürzung ihm bekannt ist.

Der Benutzer von XS-2 hat die Möglichkeit, die Befehle eines Anwendungsprogramms ohne Hilfe eines Programmierers auf seine Bedürfnisse zuzuschneiden. Er kann interaktiv im System Befehle umbenennen, in einem Menü anders gruppieren oder in Untermenüs aufspalten. Der Be-

nutzer bestimmt somit selber, wie ein von ihm gebrauchtes Werkzeug beschaffen sein soll. Dadurch lernt er gleichzeitig, Kommandostrukturen für noch nicht bestehende Applikationen zu entwickeln. Ein Skelett des von ihm gewünschten Anwendungsprogramms wird für ihn automatisch erstellt. Zusätzlich können in eine Anwendung mehrere alternative Befehlsstrukturen zu einem Befehl eingefügt werden. Das System kann damit sowohl von ungeübten als auch von erfahrenen Benutzern bedient werden.

XS-2 kann man als eine Dialogmaschine auffassen, die das Betriebssystem erweitert. Ein weiterer Aspekt einer solchen Systemstruktur ist die Definition einer Standardschnittstelle zwischen der Dialogmaschine und den Anwendungsprogrammen: Die Sorge um die konsistente Dialogbehandlung wird den Anwendungsprogrammen weitgehend abgenommen. Dies erleichtert auch die Portabilität: Es genügt, die Dialogmaschine zu übertragen, um alle Anwendungsprogramme an einen neuen Standard in der Dialogführung anzupassen. Auf diese Art wurden an der ETH Zürich Werkzeuge zur Software-Entwicklung als Anwendungen unter XS-2 entwickelt, dann auf die an den Lilith-Rechnern übliche Dialogführung angepasst und schliesslich auf Macintosh-Personalcomputer übertragen.

Adresse der Verfasserin:
Valérie Kistler
Asylstrasse 92, CH-8032 Zürich

Über die Rotation der Erde

I. Bauersima

Diese Arbeit erscheint in drei Teilen, von denen der vorliegende der erste ist. Sie befasst sich in allgemein verständlicher Form mit der Rotation der Erde als einem Problem, das mit unseren Vorstellungen über den Raum und die Zeit eng verknüpft ist. In diesem Sinne nämlich werden die astronomischen und physikalischen Aspekte der Rotation der Erde erläutert. Dies geschieht historisierend, wie aus dem Inhaltsverzeichnis ersichtlich.

Ce travail paraîtra en trois parties, dont ce qui suit constitue la première. Il traite sous une forme accessible à tous de la rotation de la terre, considérée comme un problème qui est étroitement lié à l'image que nous nous faisons de l'espace et du temps. C'est dans ce sens aussi que sont abordés les aspects astronomiques et physiques de la rotation terrestre. Le tout est décrit dans son déroulement historique, ainsi que le montre la table des matières.

Inhaltsverzeichnis

Hipparchos
Ptolemäisches Weltbild
Aristarch'sches Weltbild: Kopernikus
Tycho Brahes Mars-Beobachtungen

Kepler'sches Weltbild
Entdeckungen Galileo Galilei's
Die drei Newton'schen Bewegungsgesetze
Hamilton'sches Prinzip
Kontroverse über den absoluten Raum

(Newton, Leibnitz, Mach)
Das Newton'sche Gravitationsgesetz
Problem der Bewegung der Erde und das Schema seiner Lösung
Definition der Newton'schen Himmelsphäre
Prinzip der Richtungsbestimmung in bezug auf ein Inertialsystem
Erde als erstarnte Gleichgewichtsfigur
Die Euler'sche Theorie der Rotation eines rotationssymmetrischen Starrkörpermodells der Erde
Die Polschwankung
Unregelmässigkeiten der Rotation der Erde
Die Flutreibung

Hipparchos

1. Die Rotation der Erde ist ein Phänomen, das wir mit unseren Sinnen nicht unmittelbar empfinden. Denn die Resultante der Gravitations- und Zentrifugalkräfte ist in Bezug auf die Erdoberfläche zeitlich weitgehend konstant, so wie sie es auch sein würde, wenn die Erde nicht rotierte.