

Anforderungen und Konstruktionsprinzipien für zukünftige Benutzerschnittstellen

Bei der Untersuchung vieler Anwendungssysteme fällt auf, daß die Benutzerschnittstelle nahezu unstrukturiert mit dem eigentlichen Anwendungssystem vermischt ist (siehe Abbildung 1). Im

Beitrag von Michael Herczeg

Programm wechseln sich in aufeinanderfolgenden Programmstatements Berechnungen, Datenzugriffe und Ein-/Ausgaben beliebig ab. Dies liegt zum einen an den verwendeten Programmiersprachen (z. B. Assembler, FORTRAN, BASIC), die Strukturierung kaum unterstützen, und zum anderen an mangelnder Einsicht und geringem Interesse der Softwareentwickler, der Benutzerschnittstelle dasselbe Interesse zu widmen, wie der Funktionalität von Systemen.

Die Benutzerschnittstelle, gelegentlich auch als Mensch-Computer-Schnittstelle bezeichnet, ist die Brücke zwischen dem Benutzer und dem Anwendungssystem auf dem Computer. Durch sie wird festgelegt, wie die Kommunikation zwischen Benutzer und Anwendungssystem verlaufen soll. Dazu regelt sie Eingabemöglichkeiten des Benutzers, sowie die Ausgaben des Systems auf dem Bildschirm. Die Qualität der Benutzerschnittstelle beeinflusst entscheidend die Komplexität und Effizienz der vom Benutzer mit dem System getätigten Arbeitsabläufe. Sie bestimmt, wie stark sich der Benutzer bei seiner Arbeit an das System anzupassen hat und damit, wie stark er durch diese Arbeit belastet wird.

Diese Art der Systemkonstruktion führt zu einer ganzen Reihe von Problemen, die nicht nur die Benutzer, sondern in nicht unbedeutendem Ausmaß auch die Systementwickler belasten. Es lassen sich inzwischen eine ganze Reihe von Anforderungen definieren,

um diese Probleme zu dämpfen oder ganz zu beseitigen.

Anforderung 1:

Einfache Änderbarkeit

Durch die unstrukturierte oder konzeptionslose Realisierung von Benutzerschnittstellen ist es häufig nur sehr schwer oder praktisch sogar unmöglich nennenswerte Änderungen daran vorzunehmen. Der Benutzer hat üblicherweise keine Möglichkeit am Entwurf der Benutzerschnittstelle eines Systems in nennenswerter Weise zu partizipieren. Ist ein System erst einmal fertiggestellt, so werden Änderungen zur Beseitigung von Bedie-

nungsschwierigkeiten nur in den seltensten Fällen ausgeführt. Der Benutzer ist gezwungen sich an das System anzupassen, obwohl es oft keine Begründung für spezielles unangenehmes Verhalten des von ihm bedienten Systems gibt. Es ist anzustreben, Benutzerschnittstellen so zu konzipieren und realisieren, daß ein möglichst großer Teil von Änderungswünschen einfach und schnell berücksichtigt werden kann.

Anforderung 2:

Schnelle Erstellbarkeit

Die schnelle Erstellbarkeit von Benutzerschnittstellen hat neben wirtschaftlichen Erwägungen auch den Nutzen, daß dem Anwender schon sehr früh während der Erstellung eines neuen Systems ein Prototyp vorgestellt werden kann (*Rapid Prototyping*). Anhand solcher Prototypen kann der Anwender sehr viel besser entscheiden, ob die Lösung seinen Vorstellungen und Anforderungen entspricht, als ihm dies anhand umfangreicher textueller Beschreibungen, z. B. in Form von Pflichtenheften; möglich wäre. Durch Kritik am vorgeführten Prototypen kann er seine Wünsche und Anforderungen sehr viel leichter artikulieren.

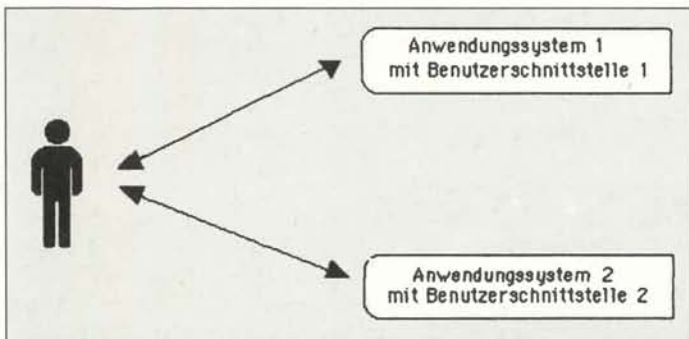


Abbildung 1: Die Benutzerschnittstelle als integraler Bestandteil der Anwendungssysteme

Konstruktionsprinzip 1: Modulare Benutzerschnittstellen

Ein entscheidender Schritt zur Erfüllung der beiden ersten Anforderungen, ist die Abkopplung der Benutzerschnittstelle vom Anwendungsteil des Systems. Zur Modularisierung der Benutzerschnittstelle isoliert man insbesondere das Ein-/Ausgabeverhalten des Systems und bildet eigenständige und leicht veränderbare Bausteine. Die Zusammenfassung dieser Bausteine kann dann selbst als Modul zum Anwendungssystem betrachtet werden (siehe Abbildung 2).

Anforderung 3: Verschiedene Benutzerschnittstellen zu einem Anwendungssystem

Es wird selten möglich sein, eine Benutzerschnittstelle zu realisieren, die den Anforderungen einer heterogenen Benutzergruppe gerecht wird. Lassen sich die Benutzer allerdings in einigermaßen homogene Untergruppen unterteilen, so lassen sich durch das Anbieten mehrerer Benutzerschnittstellen für ein und dieselbe Anwendung (siehe Abbildung 3) einige der Probleme beseitigen oder auf ein erträgliches Maß reduzieren. Eine bessere Lösung stellen adaptierbare und adaptive Benutzer-

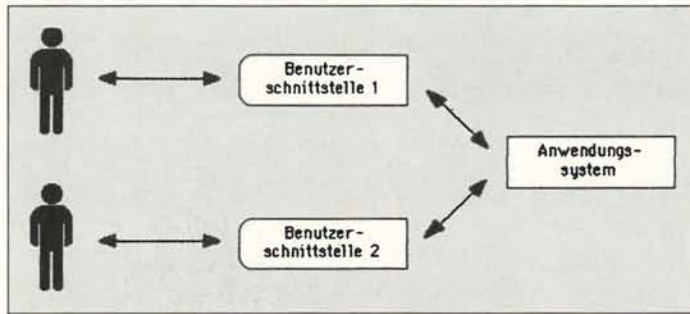


Abbildung 3: Benutzerspezifische Schnittstellen zu einem Anwendungssystem

schnittstellen dar (Anforderung 6), da sie auf individuelle Unterschiede eingehen. Sie erfordern allerdings einen ungleich höheren Erstellungsaufwand. Mehrere Benutzerschnittstellen für eine Anwendung sind auch ein wertvolles Hilfsmittel für ergonomische Untersuchungen. So läßt sich beispielsweise die Eignung verschiedener Interaktionstechniken für bestimmte Anwendungen bzw. Benutzer abschätzen (siehe Abbildung 4).

zept realisieren (virtuelle Terminals). Diese Hardwarekomponenten beeinflussen nicht primär die Funktionalität des Systems, sondern im wesentlichen seine Bedienung und sein Ausgabeverhalten. Dies gilt natürlich auch für den Übergang auf neue Terminalhardware. Diese Anforderung wird durch die Modularisierung der Benutzerschnittstelle schon teilweise gelöst. Sie erfordert allerdings eine sehr klare Trennung von

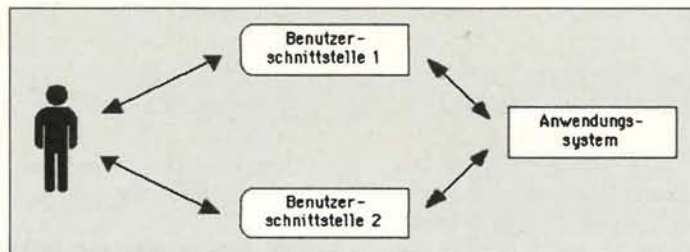


Abbildung 4: Mehrere Schnittstellen für einen Benutzer zu einem Anwendungssystem

Auch die Verwendung unterschiedlicher Bildschirme, Tastaturen und Zeigegeräte (Cursortasten, Maus, Graphiktablett,...) läßt sich über dieses Kon-

struktionsprinzip realisieren. Diese strikte Trennung ist oft schwierig zu realisieren, so daß wir dieser Forderung oft erst durch aufwendige Realisierungsmethoden nachkommen können (siehe Konstruktionsprinzip 3).

Anforderung 4: Einhaltung ergonomischer Prinzipien

Während in früheren Jahren bei Computersystemen fast

ausschließlich die Effizienz und Funktionalität Qualitätskriterien waren, werden nun zunehmend ergonomische Gesichtspunkte zum Entwurf und zur Beurteilung herangezogen. Dies sind zunächst sogenannte hardware-ergonomische Forderungen, die die Gestaltung des Bildschirmarbeitsplatzes betreffen, angefangen bei der richtigen Auswahl des Sitzplatzes über die Ausgestaltung der Tastatur bis hin zu augenschonenden Bildschirmen. Durch zum Teil schon lange vorhandenes Wissen über physiologische Eigenschaften des Menschen und durch eine Vielzahl von Experimenten, konnten inzwischen konkrete Richtlinien erstellt werden, die bereits in Normen (z. B. DIN 66233) ihren Niederschlag gefunden haben.

Sehr viel langsamer vollzog sich eine Orientierung hin zu leicht bedienbaren Anwendungssystemen. Die benutzergerechte Gestaltung von Mensch-Computer-Schnittstellen zu Anwendungssystemen ist das Thema einer relativ neuen, interdisziplinären Wissenschaft, der *Software-Ergonomie*. Auch die Software-Ergonomie tritt inzwischen mit ersten Empfehlungen zur Gestaltung der Mensch-Computer-Schnittstelle an die Software-Entwickler heran (z. B. Entwürfe zur DIN 66234). Von diesen Empfehlungen hin zu Realisierungen ist jedoch ein weiter Weg, der zunächst über experimentelle Systeme führen muß, mit denen die technische Realisierbarkeit und der Nutzen abzuschätzen sind.

Man wird davon ausgehen müssen, daß die Realisierung von Benutzerschnittstellen nicht mehr einfach vom Anwendungsprogrammierer zur Anwendung „hinzugestrickt“ werden

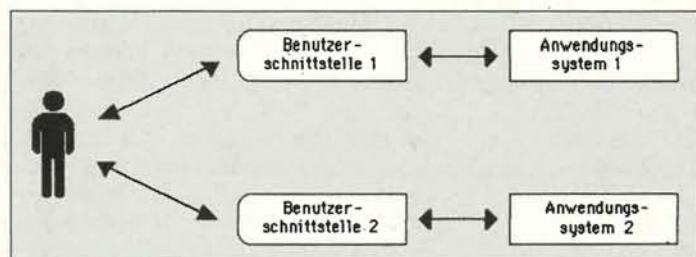


Abbildung 2: Die Benutzerschnittstelle als Modul zum Anwendungssystem

kann, sondern von Spezialisten durchzuführen ist, die außer über Programmierwissen auch über Kenntnisse ergonomischer Rahmenbedingungen verfügen. Diese Vorgehensweise setzt eine von der Anwendungsentwicklung teilweise losgelöste Entwicklung der Benutzerschnittstelle voraus.

Anforderung 5:

Konsistenz bei Benutzung mehrerer Systeme

Häufig sind durch einen Benutzer mehrere Systeme zu bedienen. Dies darf nicht dazu führen, daß mit jedem neuen System neue Interaktionstechniken erlernt werden müssen. Nur durch die Einführung integrierter Systeme, die dem Benutzer eine homogene Schnittstelle zu allen seinen Anwendungen bieten, wird ihm unbelastetes Arbeiten möglich sein. Diese *Konsistenz in der Bedienung* erfordert aber allgemeine, möglichst anwendungsneutrale Interaktionsformen, wie beispielsweise Menüs. Es gilt nun, neue derartige Interaktionsformen zu entwickeln und erproben.

Konstruktionsprinzip 2: Anwendungsneutrale Benutzerschnittstellen

Wird für jedes Anwendungssystem eine neue Benutzerschnittstelle erstellt, so bedeutet dies natürlich auch einen beträchtlichen Zeit- und Kostenaufwand für die Erstellung und Wartung. Der Nutzen der Modularisierung wird besonders groß, wenn ein Benutzerschnittstellenmodul in derselben oder gering individualisierten Form für mehrere Anwendungssysteme benutzt werden kann (siehe

Abbildung 5). Man spricht in diesem Zusammenhang von *anwendungsneutralen Benutzerschnittstellen* (Dzida 83). Dies ist eine mögliche Konsequenz aus den beiden letztgenannten Anforderungen.

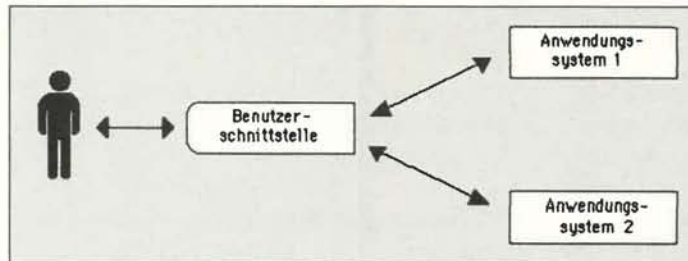


Abbildung 5: Anwendungsneutrale Benutzerschnittstelle

Die Entwicklung von Masken- und Menügeneratoren für viele kommerzielle Anwendungen war bereits ein Schritt in diese Richtung. Diese Generatoren sind ein heutzutage übliches Werkzeug vieler Software-Ersteller, um standardisierte Benutzerschnittstellen (in diesem Fall Menü- und Formulardialoge) schnell und weitgehend konsistent für nahezu beliebige Anwendungen zu erstellen.

Anforderung 6:

Adaptierbare und adaptive Benutzerschnittstellen

Es gibt nicht „den“ Benutzer eines Systems. Jeder Benutzer hat seine eigenen Erfahrungen und Kenntnisse in der Bedienung von Computersystemen. Viele Benutzer entwickeln sich erst während der Arbeit mit einem neuen System vom Computereiniger zum erfahrenen Benutzer. Die wenigsten heute existierenden Computersysteme passen sich der Entwicklung des Benutzers an. Dies führt dazu, daß Benutzer am Anfang überfordert sind; im Lauf der Zeit stoßen sie jedoch an die Grenzen der Möglichkeiten der Systeme und

fühlen sich eingeschränkt. Benutzer unterscheiden in ihren sensorischen und motorischen Fähigkeiten. Sie haben darüber hinaus persönliche Vorlieben. Dies kann von der oberflächlichen Gestaltung des Systems (z. B. Layout auf dem

Bildschirm, Farbgebung, Hervorhebungen, akustische Signale) bis zur Auswahl von Interaktionsmethoden (z. B. Menü- oder Kommando-Interaktion) reichen.

Was benötigt wird, sind *adaptive* (sich an den Benutzer anpassende) und *adaptierbare* (durch den Benutzer anpassbare) Systeme. Die Systeme müssen zu diesem Zweck, unter anderem, Modelle von ihren verschiedenen Benutzern halten (Wissen über die Benutzer).

Anforderung 7:

Verfeinerte Interaktions- und Visualisierungsmethoden

Anwendungssysteme werden komplexer, da immer schwierigere Aufgaben damit bewältigt werden. Die erhöhte Komplexität schlägt sich in der Benutzerschnittstelle nieder. Dem Benutzer wird eine höhere Funktionalität angeboten, die Darstellungen auf den Bildschirmen werden umfangreicher, die Lösung von Aufgaben erfordert das häufige Wechseln zwischen verschiedenen Kontexten. Die herkömmlichen Verfah-

ren zur Gestaltung von Benutzerschnittstellen, z. B. die Verwendung ganzseitiger Bildschirmmasken und Menüs, sind diesen neuen Bedingungen nicht gewachsen. Ansätze wie Fenstersysteme, „Pop-Up-Menüs“, ikonische Darstellungen und die Verwendung eines Zeigeelements (z. B. Maus) machen veränderte Programmierstechniken sinnvoll. Es bieten sich vielfältige Methoden zur Wissensrepräsentation an, z. B. objektorientierte Programmierung.

Konstruktions-

prinzip 3: Wissensbasierte Benutzerschnittstellen

Wir erwarten von zukünftigen Benutzerschnittstellen Eigenschaften wie kooperatives und intelligentes Verhalten, das an den Benutzer und seine Tätigkeit angepaßt ist. Um dies zu gewährleisten, muß in den Benutzerschnittstellen vielfältiges Wissen repräsentiert und verfügbar gemacht werden. Das Resultat sind sogenannte *wissensbasierte Benutzerschnittstellen*. Entwirft man die Benutzerschnittstelle als wissensbasiertes System (siehe Abbildung 6), so muß zunächst untersucht werden, welches Wissen benötigt wird und repräsentiert werden soll. Nachfolgend wird abstrahiert aufgelistet, welches die wichtigsten Informationen sind, mit denen eine solche Benutzerschnittstelle umzugehen hat:

● Wissen über Bildschirm, Tastatur und Zeigeelemente

Da die Gegebenheiten der Kommunikationshardware häufig ausschlaggebend für die Mensch-Computer-Kommunikation sind, müssen diese dem Sy-

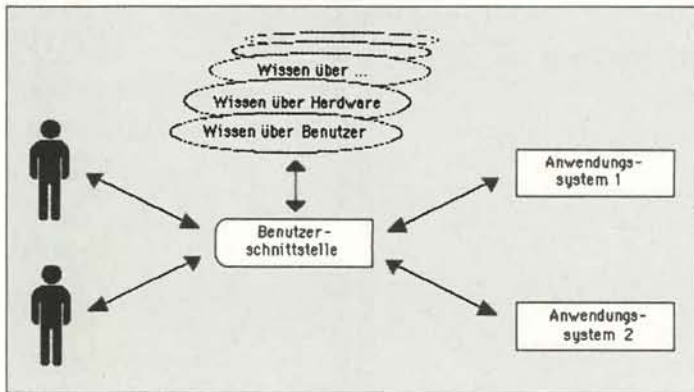


Abbildung 6: Wissensbasierte Benutzerschnittstelle

stem bekannt sein. Kennt das System beispielsweise die Fähigkeiten des Bildschirms, so kann es die Ausgaben an den Benutzer in Umfang und Form variieren. Auf einem Farbbildschirm werden Hervorhebungen dann durch eine auffällige Farbe bewerkstelligt, während sie auf einem monochromen Bildschirm durch Inversdarstellung realisiert werden können.

● Wissen über Anwendungssysteme

Damit eine Benutzerschnittstelle sinnvoll zwischen Benutzer und Anwendungssystem agieren kann, ist es notwendig, sie mit Information über das Anwendungssystem zu versorgen. Beispielsweise kann sie wissen, in welchem Format eine bestimmte Eingabe des Benutzers an das Anwendungssystem weiterzuleiten ist. Auf der anderen Seite kann sie Ausgabeinformationen vom Anwendungssystem geeignet aufbereiten, bevor sie auf dem Bildschirm dargestellt werden.

● Wissen über ergonomische Anforderungen

Wie eingangs erwähnt, ist es wichtig, die ergonomischen Forderungen zu berücksichtigen. Die

Benutzerschnittstelle kann, wenn sie Wissen über die Wirkung von Farben hat, entscheiden, ob eine bestimmte Schriftfarbe auf einem bestimmten Bildschirmhintergrund gut oder schlecht lesbar ist und ggf. eine geeignetere Auswahl treffen.

● Wissen über den Benutzer (Benutzermodell)

Der Benutzer kann durch ein System besser unterstützt werden, wenn das System Wissen über ihn hat, z. B.:
– Erfahrungsstand in der Systembenutzung
– Vorlieben für bestimmte Interaktionsformen (z. B. Menüs oder Kommandos)
– typische Schreibfehler
– Defaultwerte (z. B. immer wiederkehrende Eingabewerte beim Ausfüllen eines Bildschirmformulars)

Ein Benutzermodell ist Grundlage für ein *adaptives System*, daß sich dem jeweiligen Benutzer anpaßt.

● Wissen über die Dialoggeschichte

Die Benutzerschnittstelle kann sich den Ablauf des Dialogs merken (Dialoggeschichte), um den Benutzer durch *UNDO-Funktionen* (Stornieren bereits ausgeführter Aktionen), *REDO-Funktionen* (Wiederholen

bereits ausgeführter Aktionen) und durch *Erklärung der Dialoggeschichte* zu unterstützen.

● Selbstwissen

Soll dem Benutzer Hilfe in der Benutzung des Systems gegeben werden, so muß unter anderem auch Wissen über die Benutzerschnittstelle selbst repräsentiert werden. Man könnte dies als das Selbstwissen des Systems bezeichnen.

Erklärt das System die Benutzerschnittstelle nicht nur, sondern gibt dem Benutzer gleichzeitig die Möglichkeit diese zu verändern, so erhalten wir ein *adaptierbares System*.

Fassen wir die Grundprinzipien noch einmal zusammen, denen bei der Konstruktion von zukünftigen Benutzerschnittstellen Rechnung getragen werden sollte:

1. Benutzerschnittstellen sollten *modular* aufgebaut,
2. so weit wie möglich *anwendungsneutral* und
3. *wissensbasiert* sein.

Diese Prinzipien ergeben sich aus den eingangs diskutierten Forderungen an zukünftige Benutzerschnittstellen. Die Umsetzung in nutzbare Systeme wird eine anspruchsvolle Aufgabe in den kommenden Jahren sein. Es gibt allerdings schon heute vielversprechende Ansätze, mit denen man experimentieren muß und die es zu verbessern und zu erweitern gilt. Ein solcher Ansatz ist die *objektorientierte Implementierung von Benutzerschnittstellen*, an der wir in unserem Forschungsprojekt derzeit arbeiten. Unsere bisherigen Erfahrungen sprechen für diesen Ansatz, weitere Experimente sollen dies noch vertieft nachweisen.

Literatur zum Thema „Benutzerschnittstellen“

- (Bauer et al. 82)
J. Bauer, H.-D. Böcker, F. Fabian, G. Fischer, R. Gunzenhäuser, C. Rathke: „Entwicklung von Methoden und Werkzeugen zur Verbesserung der Mensch-Maschine-Kommunikation“. Antrag an das BMFT, Institut für Informatik, Universität Stuttgart, 1982
- (Bauer, Schneider 85)
J. Bauer, M. Schneider: „Hilfe- und Dokumentationssysteme im Softwareengineeringprozeß“. In ONLINE '85. 8. Europäische Kongressmesse für Technische Kommunikation, pp 70-1-70-18. Düsseldorf, Februar, 1985
- (Dzida 83)
W. Dzida: „Das IFIP-Modell für Benutzerschnittstellen“. Office Management 31 (Sonderheft), pp 6-8, April, 1983.
- (Goldberg 81)
A. Goldberg (ed.): „SMALLTALK“. BYTE 6(8), 1981. Special Issue.
- (Goldberg, Robson 83)
A. Goldberg, D. Robson: „SMALLTALK-80, The Language and its Implementation“. Addison-Wesley, Reading, Ma., 1983.
- (Hayes 84)
P. J. Hayes: „Executable Interface Definitions using Form-Based Interface Abstractions“. Technical Report, Carnegie-Mellon University, Computer Science Department, Pittsburgh, April, 1984.
- (Hayes, Szekely, Lerner 85)
P. J. Hayes, P. A. Szekely, R. A. Lerner: „Design Alternatives for User Interface Management Systems Based on Experience with COUSIN“. In L. Borman, B. Curtis (editor), *Human Factors in Computing Systems*, pp 169-175. CHI-85 Conference Proceedings, New York, April, 1985.
- (Herczeg 83)
M. Herczeg: „DYNAFORM – Ein interaktives Formularsystem zum Aufbau und zur Bearbeitung von Datenbasen“. In H. Balzert (editor), *Software-Ergonomie*, pp 135-146. German Chapter of the ACM, Stuttgart, 1983.
- (Herczeg et al. 85)
M. Herczeg, D. Maier, C. Rathke, W.-F. Riekert: „Vom Dialogsystem zur wissensbasierten Mensch-Computer-Kommunikation“. In ONLINE '85. 8. Europäische Kongressmesse für Technische Kommunikation, pp 2P-1-2P-14. Düsseldorf, Februar, 1985.
- (Laubsch, Rathke 83)
J. Laubsch, C. Rathke: „OBJTALK: Eine Erweiterung von LISP zum objektorientierten Programmieren“. In H. Stoyan, H. Wedekind (editors), *Objektorientierte Software- und Hardwarearchitekturen*, pp 60-75. Stuttgart, 1983.
- (Multiplan 84)
Microsoft: „Microsoft Multiplan“. Microsoft Corporation, Bellevue, WA, 1984. User Manual for the Apple Macintosh
- (Rathke 84)
C. Rathke: „ObjTalk Primer“. Institutsbericht, Institut für Informatik, Universität Stuttgart, 1984
- (Smith et al. 82)
D. C. Smith, Ch. Irby, R. Kimball, B. Verplank: „Designing the Star User Interface“. BYTE, April 1982

Die Forschungsgruppe INFORM

Die Forschungsgruppe INFORM am Institut für Informatik der Universität Stuttgart erarbeitet Grundlagen für den Entwurf und die Realisierung wissensbasierter Systeme (WBS) sowie moderner Techniken der Mensch-Computer-Kommunikation (MCK). In der Forschungsgruppe arbeiten derzeit 10 wissenschaftliche Mitarbeiter unter der Leitung von Prof. Dr. R. Gunzenhäuser.

Die Arbeiten von INFORM basieren auf Forschungsarbeiten der Künstlichen Intelligenz (KI) und der Kognitionswissenschaft („Cognitive Science“). Repräsentation von Wissen, Verstehen der kognitiven Fähigkeiten des Menschen, Konzeption anwendungsneutraler Benutzerschnittstellen und Unterstützung explorativen Programmierens sind einige Forschungsgebiete, die eine wichtige Rolle in der Arbeit der Gruppe spielen.

Ziel der Forschung ist nicht der Entwurf vollautomatischer Systeme, sondern die Entwicklung von Systemen, die durch eine sinnvolle Arbeitsteilung sowie reichhaltige Kommunikationsmöglichkeiten zwischen Mensch und Computer gekennzeichnet sind. Ein langfristiges Forschungsziel ist es, dem Benutzer die Möglichkeit zu geben, die Art der Interaktion und die verwendeten Werkzeuge nach seinen eigenen Vorstellungen und Wünschen zu gestalten.

Diese Arbeiten werden seit 1981 vom Bundesministerium für Forschung und Technologie in drei aufeinanderfolgenden Forschungsprojekten gefördert.

INFORM-81 (Die Funktion von integrierten Informationsmanipulationssystemen [IMS]) zur Unterstützung der Mensch-Maschine-Kommunikation) hatte den Entwurf und die Implementation eines IMS zum Ziel, bei dem die Benutzer alle Arbeiten (z. B. Design-Spezifikationen, Pläne, Dokumentationen, Berichte, elektronische Post, Dokumentenablage, Formulare, Terminplanung usw.) mit Hilfe eines Computersystems mit einheitlicher Mensch-Computer-Schnittstelle verrichten können.

Wissensbasierte Systeme erschließen in der Mensch-Computer-Kommunikation qualitativ neue Möglichkeiten. Ziel des Projekts INFORM-83 war es, benutzerorientierte Computersysteme zu definieren, zu spezifizieren und zu implementieren, die wissensbasierte Ansätze in der MCK realisieren. Zu den in INFORM-83 bearbeiteten Aufgaben gehörten die Implementierung wissensbasierter Schnittstellenkomponenten und deren Einsatz in Anwendungssystemen aus dem Bereich der Büroautomatisierung und der Software-Produktion.

Seit 1. 1. 1985 beteiligt sich die Forschungsgruppe INFORM am Verbundprojekt WISDOM (Wissensbasierte Systeme zur Bürokommunikation: Dokumentenbearbeitung, Organisation, Mensch-Computer-Kommunikation). In diesem Projekt kooperiert INFORM u. a. mit der Firma Triumph-Adler, der Gesellschaft für Mathematik und Datenverarbeitung (GMD) und dem Institut für Arbeitswirtschaft und Organisation der Fraunhofer-Gesellschaft

(FhG) in Stuttgart. Projektziel ist die Entwicklung prototypischer, wissensbasierter Systeme, die dem Benutzer im Bürobereich umfangreiche Unterstützung anbieten. Die Arbeiten der Forschungsgruppe konzentrieren sich in diesem Projekt auf folgende Bereiche:

– *anwendungsunabhängige MCK*

Die Realisierung von anwendungsunabhängig einsetzbaren Komponenten erleichtert die Konstruktion der Mensch-Computer-Schnittstelle für Anwendungssysteme und ermöglicht eine konsistente Schnittstellengestaltung für komplexe rechnerunterstützte Arbeitsumgebungen.

– *Grundlagen aktiver und passiver Hilfesysteme*

Ausgehend von bereits vorhandenen Hilfefunktionen werden neue Konzepte zur Akquisition und Darbietung von Unterstützungsinformation entwickelt und in prototypischen Systemen erprobt.

– *Wissensrepräsentation*

Ziel der Arbeit in diesem Bereich ist die Schaffung neuer Konzepte zur Repräsentation komplexer Wissensstrukturen über Dokumente und Vorgänge im Bürobereich sowie Entwurf und Realisierung einer Programmierungsumgebung für eine objektorientierte Wissensrepräsentationssprache.

– *Objektorientierte Modellierung und Akquisition von Wissen*

Metasysteme ermöglichen dem Endbenutzer die Erzeugung und Modifikation von Wissensentitäten, Regeln und Kon-

zepten zur Repräsentation seines Fachwissens. Mit ihrer Hilfe lassen sich Anwendungssysteme ohne Programmierkenntnisse umgestalten.

– *Systeme zur Planungsunterstützung*

Maschinelle Unterstützung bei der Planung komplexer Bürovorgänge macht es notwendig, dem Rechner Wissen über Planungsheuristiken und -strategien verfügbar zu machen. Mit diesem Wissen soll die zeitliche und inhaltliche Planung von Ablaufprozeduren ermöglicht werden.

Zu den bei INFORM entstandenen Basissystemen gehören beispielsweise die Wissensrepräsentationssprache ObjTalk, ein Fenstersystem sowie der bildschirmunabhängige Editor BISO. Daraus entstanden Werkzeuge für anwendungsunabhängige Mensch-Computer-Kommunikation (z. B. Icons, Menüs, Formulare). Mit diesen Komponenten wurden eine Softwareproduktionsumgebung für die Programmiersprache LISP (mit Dokumentationssystem, Analysekomponente usw.) sowie eine Reihe wissensbasierter Anwendungssysteme (FINANZ, PLANER u. a.) entwickelt. Im Bereich Hilfesysteme wurden die zwei Prototypen AKTIVIST und PASSIVIST realisiert.

Die Ergebnisse der Forschungsarbeit von INFORM wurden in zahlreichen Konferenzbeiträgen, Systemvorführungen, Workshops und Veröffentlichungen der Fachwelt zugänglich gemacht.

Einzelpreis DM 10,-

&SoftwareMagazin

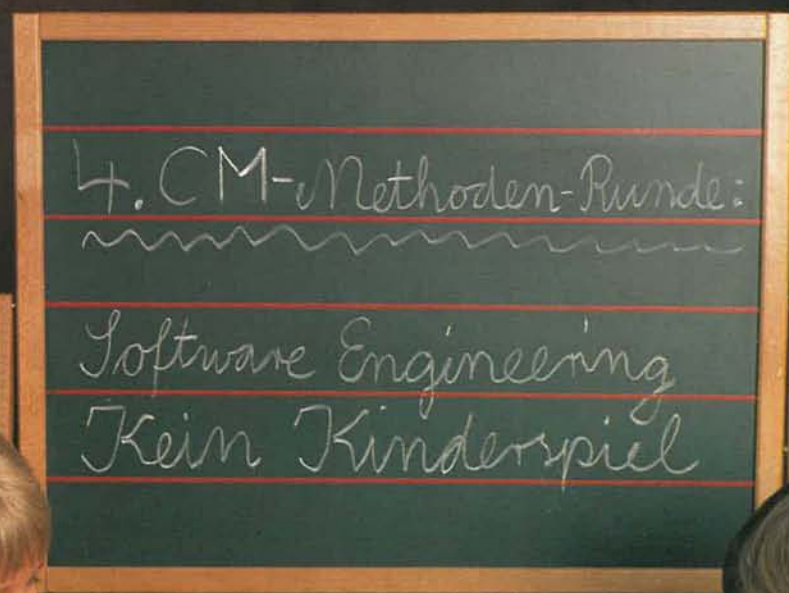
computer

magazin

D 1518 E

Oktober 1985, 14. Jahrgang

10'85



Gebühr bezahlt

Postvertriebsstück

