

# Software- Ergonomie im Produktionsbereich

## Dargestellt am Beispiel der Analyse und Gestaltung von Programmiersystemen für Werkzeugmaschinen

H.-J. Bullinger, C. Raether, K.-P. Fährlich, M.Kärcher

**Zusammenfassung:** Fragestellungen aus dem Produktionsbereich sind momentan innerhalb der Software-Ergonomie unterrepräsentiert. Für diesen Bereich mit hohem Zuwachs an Bildschirmarbeitsplätzen werden Analyse- und Gestaltungsarbeiten zu Programmierumgebungen an CNC-Maschinen vorgestellt. Diskutiert werden dabei Software-Werkzeuge, die zur effizienten Implementation software-ergonomisch gut gestalteter Benutzerschnittstellen herangezogen werden können.

### 1 Problemstellung

Für Jahrzehnte lag der Schwerpunkt ergonomischer **Analyse, Gestaltungs- und Evaluationsarbeiten** im Produktionsbereich. Im letzten Jahrzehnt wuchsen durch den Einsatz von Informations- und Kommunikationssystemen im Bürobereich neue Arbeitsschwerpunkte hinzu. Aufgrund der Beobachtung der Autoren muß festgestellt werden, daß der Software-Ergonomie im Produktionsbereich, bezogen auf ihre Relevanz, momentan zu wenig Aufmerksamkeit gewidmet wird.

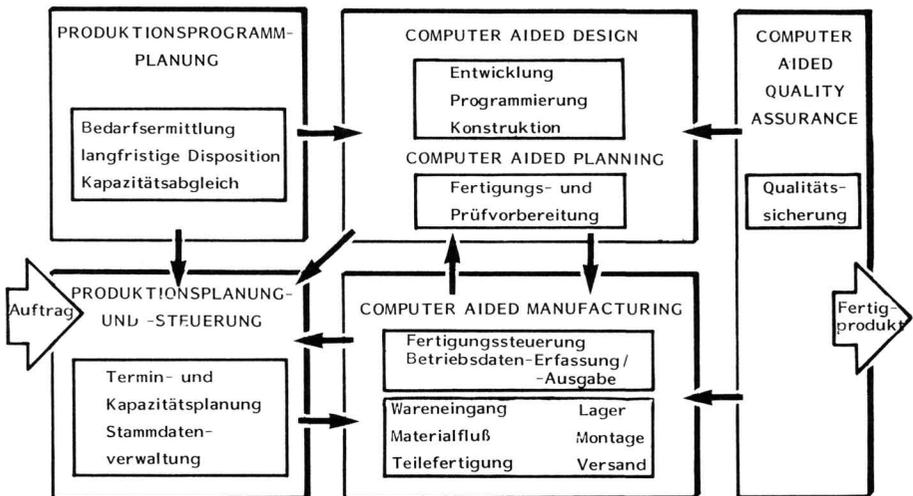


Bild 1: CIM: Computerunterstützte Produktion

Die Arbeiten basieren auf folgenden Projekten: Graphischer Editor für Drehbearbeitung, Interaktive Maschinendiagnose, Fertigungsleitwarte (Industrie); Fertigungsorientierte Programmierverfahren (PFT). Neben den Autoren haben mitgearbeitet: O. Böhm, S. Braun, M. Dannenberg, M. Eckerle, A. Luik, H. Meitner, V. Schwarz, M. Thines, B. Tille, A. Vogel, J. Wagner, J. Walser, G. Wasserlos.

Das Unternehmen der Zukunft ist charakterisiert durch die informationstechnische Durchdringung und Vernetzung aller wesentlichen Unternehmensbereiche. In allen diesen Bereichen kommen in Zukunft massiv DV-Technologien mit interaktiven Bildschirmarbeitsplätzen zum Einsatz. Das folgende Bild zeigt für eine Stichprobe (n = 84) deutscher Unternehmen die Zunahme der eingesetzten Systeme für einen Zeitraum von 7 Jahren (1980-1987), aufgliedert in die drei wichtigsten Bereiche Konstruktion, Arbeitsvorbereitung und Fertigung:

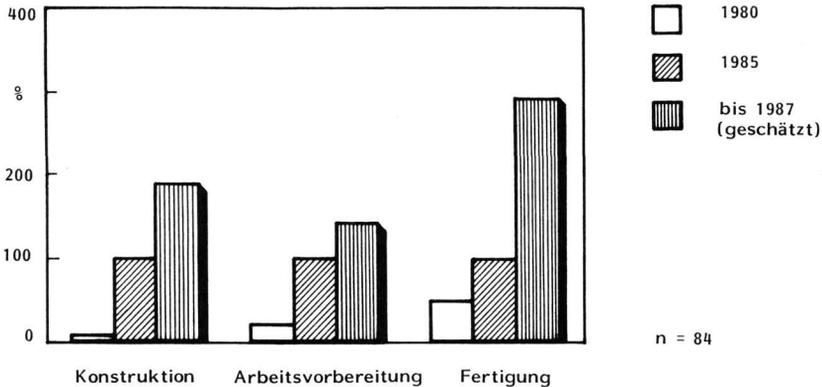


Bild 2: Zunahme von Bildschirmarbeitsplätzen in ausgewählten Produktionsbereichen

Die Daten deuten darauf hin, daß. in den untersuchten Bereichen weiterhin mit hohen Zuwachsraten zu rechnen ist. In besonderem Maße wird jedoch für den Fertigungsbereich (CNC-Maschinen, Fertigungsleitstände, Robotersteuerungen etc.) ein extremer Anstieg ( von 100% im Jahre 1985 auf ca. 290% im Jahre 1987) prognostiziert. Absolut gesehen und bezogen auf die Anzahl betroffener Mitarbeiter, ist der Durchdringungsgrad in den Konstruktionsabteilungen mittlerweile am höchsten - ein Indikator, daß diesem Bereich aus software-ergonomischer Sicht in Zukunft verstärkte Aufmerksamkeit gewidmet werden sollte.

## 2 Gestaltungsdefizite herkömmlicher CNC- Systeme aus arbeitswissenschaftlicher Sicht

Im Laufe der letzten beiden Jahre wurden Gestaltungsdefizite im Bereich der CNC-Systeme aus arbeitswissenschaftlicher Sicht aufgearbeitet. Die zentralen Aussagen dazu lauten /1/, /2/:

- o Die CNC- Programmierung stellt sich im Werkstattbereich - obwohl von den Zeitanteilen her nicht tätigkeitsbestimmend - inhaltlich als Engpaß heraus.
- o Facharbeiter und angelernte Maschinenbediener sind konkret gegenständliches Arbeiten und graphische Repräsentation aufgrund ihrer Ausbildung gewohnt - eine Denk- und Arbeitsweise in Visualisierungen liegt ihnen näher als in Algorithmen bzw. abstrakten Programmier- oder Kommandosprachen.
- o Bei realisierten Systemen wurde diese Arbeits- und Denkweise nur unzureichend unterstützt. Es bestand daher ein starker Trend zur Verlagerung höherwertiger Tätigkeitsselemente in vorgelagerte Bereiche (Arbeitsvorbereitung, Planungsabteilung).

### 3 Verbundvorhaben zur Schaffung neuartiger Programmierumgebungen in der Produktion

Die Notwendigkeit der Schaffung software-ergonomisch verbesserter Systeme wird neben der arbeitswissenschaftlichen Argumentation auch durch technische Aspekte untermauert:

- o Die Implementation von Fenstersystemen unter Verwendung komfortabler Dialogtechniken ermöglicht in höherem Maße als bei konventionellen Systemen die **Funktionsintegration** unterschiedlicher Teilkomponenten. So kann durch die Möglichkeit der gleichzeitigen Beobachtung und Beeinflußung unterschiedlicher Prozesse die Zusammenlegung von Mehrmaschinenbedienung, Parallelprogrammierung, Bedienung von Betriebsdatenerfassung- und Planungsfunktionen erfolgen.
- o Kompatibilitätsbetrachtungen weisen auf die Notwendigkeit der **Standardisierung** an der Benutzerschnittstelle hin: Einheitliche Programmeingabe in den Bereichen Werkstatt, Arbeitsvorbereitung, Planungsabteilungen, sowie bei den verschiedenen Bearbeitungsverfahren (Drehen, Fräsen, Nippeln etc.) wird angestrebt.
- o Die Möglichkeit der Implementation unterschiedlicher **Vernetzungskonzepte** (zentrale, dezentrale Lösungen) wird ebenfalls durch die geeignete Ausprägung der Benutzerschnittstelle beeinflußt.

Unter Förderung des Bundesministeriums für Forschung und Technologie wird zur Zeit in einem nationalen Verbundvorhaben mit 22 beteiligten Firmen und Forschungsinstituten das Projekt: "Werkstatorientierte Programmierverfahren" bearbeitet. Ziel des Vorhabens ist es, durch die Schaffung einer geeigneten Programmierumgebung die Erfahrung des Facharbeiters beim Einsatz moderner Fertigungseinrichtungen zu nutzen, um so auch betriebswirtschaftliche Zielgrößen wie Flexibilität, Grad der Arbeitsteilung etc. zu optimieren.

### 4. Pflichtenhefte zur Definition der Programmierumgebung an CNC- Maschinen

Die zentralen Arbeitspakete in oben genanntem Projekt, die sich mit der Gestaltung der Mensch-Computer- Interaktion befassen, werden vom Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO), Stuttgart ausgeführt. Die Software soll dabei entsprechend einem geschichteten Modell /3/, /4/, /5/ implementiert werden. Daraus abgeleitet wurde eine Gestaltungssystematik /6/. Hierbei ist ein wohldefinierter Prozeß bzgl. den Schichten: Organisationsschnittstelle, Werkzeugschnittstelle, Dialogschnittstelle und Ein-/Ausgabeschnittstelle zu durchlaufen.

**Organisationsschnittstelle** : Aufgrund der zukünftigen informationstechnischen Vernetzung und weiteren Automatisierung im Produktionsbereich werden zunehmend Daten aus anderen Bereichen des CAD/CAM- Prozesses für die CNC-Programmierung bereitgestellt. Die damit verbundene Gefahr der Verlagerung von Programmieraufgaben in andere Unternehmensbereiche (z.B. Konstruktion) kann durch eine softwareergonomisch gestaltete Benutzerschnittstelle, die die Integration der Daten Vor-Ort ermöglicht, eingeschränkt werden. Die Integration von unterschiedlichen Daten, die Herstellung von Kompatibilität über die verschiedenen Orte der Programmerstellung (Werkstatt, Arbeitsvorbereitung, Konstruktion), zwischen den einzelnen Technologieverfahren (Bohren, Fräsen, Drehen, Stanzen etc.) und nicht zuletzt zwischen den einzelnen Herstellern

erfordert die Unterscheidung der Gesamtaufgabe in einen **definitorischen** und einen **verfahrensspezifischen** Teil.

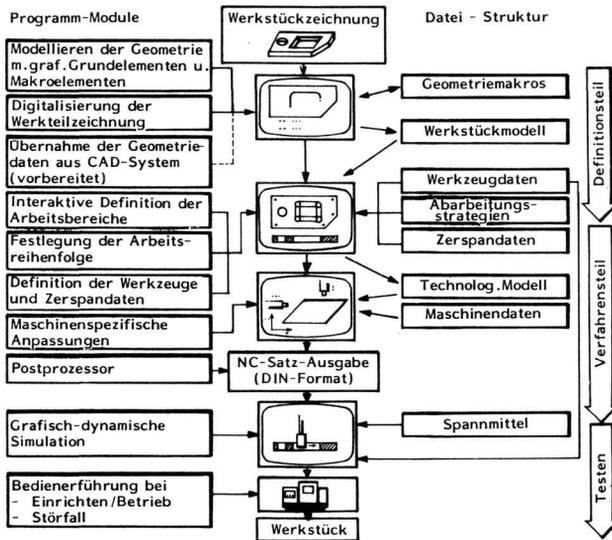


Bild 3: Verbundvorhaben "Werkstatorientierte Programmierverfahren"

Zu dem **definitorischen** Teil werden die Beschreibung der Rohteil-Geometrie, des Rohteil-Werkstoffes, der Fertigteilgeometrie, der Spansituation sowie die Bearbeitungsvorgänge gerechnet. Ferner gehört zu diesem Teil die Anlage und Pflege von Dateien (Teileprogrammdatei, Datei von Geometriemakros, Rohteildatei, Werkstoffdatei, Werkzeugdatei, Spannmitteldatei sowie die Datei zur Abspeicherung von Bearbeitungsmethoden).

Im **verfahrensspezifischen** Teil erfolgt die Festlegung der technologischen Kenngrößen - soweit nicht schon systemseitig generiert und durch den Bediener lediglich zu quittieren -, wie z.B. die Definition der Werkzeuge sowie deren Anordnung auf den Werkzeugspeicher und die Beschreibung der prozeduralen Abläufe. Die Simulation des Programmes sowie die Berechnung von Haupt- und Nebenzeiten schließen diesen Teil ab.

Für eine Vielzahl von Funktionen wurde neben der manuellen Eingabe die automatische Erzeugung der entsprechenden Parameter definiert: Dies sind u.a. Schnittwertbestimmungen, Kollisionsbetrachtungen, Vorschläge zu Bearbeitungsfolgen und Werkzeugauswahl.

**Werkzeugschnittstelle** : Ziel war es hier, eine an den spanenden Fertigungsverfahren orientierte Programmiermethode zu definieren. Aufgrund der Aufgaben- und Benutzeranalysen wurde eine, graphikorientierte Methode gewählt. Neben der Definition und Implementation eines bildschirmorientierten Graphikeditors wurden weitere graphisch-interaktive Werkzeuge zur Simulation des Bearbeitungsprozesses geschaffen.\*

\*) Die dynamischen Simulationswerkzeuge wurden vom Fraunhofer- Institut IPK, Berlin erstellt.

Realisiert wurde ferner die in diesem Kontext nicht triviale Forderung der Optimierung und Korrektur der Programme in gleicher Weise wie bei der Programmerstellung.

Für die Beschreibung der Objekt-/Funktionsbeziehungen wurden formalisierte Vorgehensweisen herangezogen /6/. Es wurde darauf geachtet, die Gesamtfunktionalität des Systemes auf möglichst "generische" Objekte und Operationen zu beziehen. Diese Vorgehensweise ist besonders dazu geeignet ,schon in einem recht frühen Stadium des Systemdesigns Aufschluß über mögliche nicht adequate Objekt- und Funktionsmengen sowie Inkonsistenzen (Bereiche mit sehr heterogener Verteilung der erlaubten Objekt-/Funktionsbeziehungen) zu erhalten.

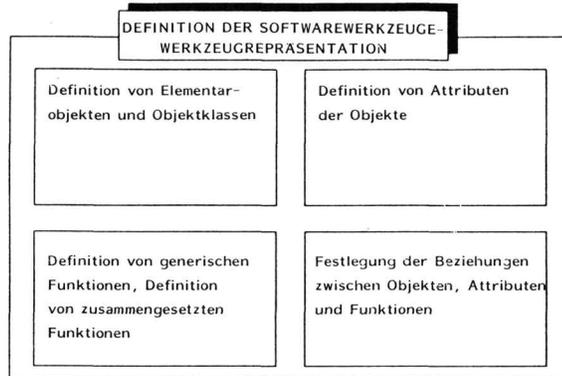


Bild 4: Werkzeugschnittstelle

**Dialogschnittstelle:** Die Dialogschnittstelle wurde in Anlehnung an die z.Z. in der Normung diskutierten Kriterien /7/ festgelegt. Dabei wurde besonderer Wert auf **Selbsterklärungsfähigkeit**, **Erlernbarkeit** und **Problemangemessenheit** gelegt. Um sowohl ungeübten als auch hochtrainierten Benutzern in angemessener Weise das Arbeiten an dem System zu ermöglichen, wurde eine benutzerinitiierte und auch gemischt rechnerinitiierte/hybride Dialogform definiert. Ein Wechsel der Initiativenlage des Dialoges ist zu jedem Zeitpunkt möglich.

Als hybride Dialogform wird das Prinzip der Direkten Manipulation /3/, /4/ angestrebt. In der momentanen Ausbaustufe des Systems sind allerdings noch Anteile reiner Menüsysteme zu finden. Die implementierten Dialogtechniken sind:

- o Funktionstastendialog für die Eingabe generell verfügbarer, kontextunabhängiger Funktionen wie z.B. Löschfunktion, Ebenrücksprung, UNDO etc.
- o Softkeymenüauswahl zur Verzweigung weiterer Dialogebenen und Eingabe der nur lokal verwendeten Funktionalität.
- o Formulareingabe zur Festlegung der entsprechenden Parameterwerte.
- o Direkte Manipulation für die Bearbeitung der Graphik, die Aufteilung der Abspanzyklen auf die Arbeitsstationen sowie die Anordnung der Werkzeuge auf dem Speicher.

Die Festlegung der Komplexität und Abfolge einzelner Dialogschritte und deren Implementation erfolgte unter Einsatz von Transitionsnetzen. Auf diese Weise war es z.B. möglich, erlaubte Sprung- und Wiederaufsetzpunkte einfach festlegen zu können.

Die Betrachtung der Dialogsyntax über die verschiedenen zum Einsatz gelangenden Dialogtechniken hinweg zeigt, daß Selektionsvorgänge den häufigsten Fall möglicher Interaktionsschritte darstellen.

**Ein-/Ausgabeschnittstelle:** Für die Ein-/Ausgabe wurden textuelle und graphische Modi gewählt. Die Daten werden dabei in eine einheitliche Repräsentation für die darüberliegenden Schichten überführt, die Einbindung weiterer E/A- Modi wie z.B. Sprache ist einfach durchführbar.

Die Eingabe der Daten erfolgt aufgrund der besonderen Umweltbedingungen im Werkstattbereich über besonders geschützte Tastaturen. Für eine spätere Ausbaustufe sind frei positionierbare Zeigeinstrumente auch für diesen Anwendungsbereich vorgesehen. Die Ausgabe der Information erfolgt über einen hochauflösenden, acht Farben darstellenden Vektorbildschirm. Der Graphikaufbau ist durch die entsprechende Hardwarekonfiguration (68000 Prozessor, Bitsliceprozessor) extrem schnell.

Bei der Werkzeugschnittstelle wurde eine Aufteilung in Elementarobjekte bzw. Objektklassen vorgenommen. An der E/A-Schnittstelle wird diese Vorgehensweise durch die Zuordnung der entsprechenden Gestaltungsparameter nach folgendem Schema komplettiert dargestellt am Beispiel der Farbzuordnung:

- a) Bestimmung von Elementarobjekten: Größte Informationseinheiten, die bezüglich aller Gestaltungsparameter nicht weiter zu unterschieden werden brauchen, sind festzulegen:

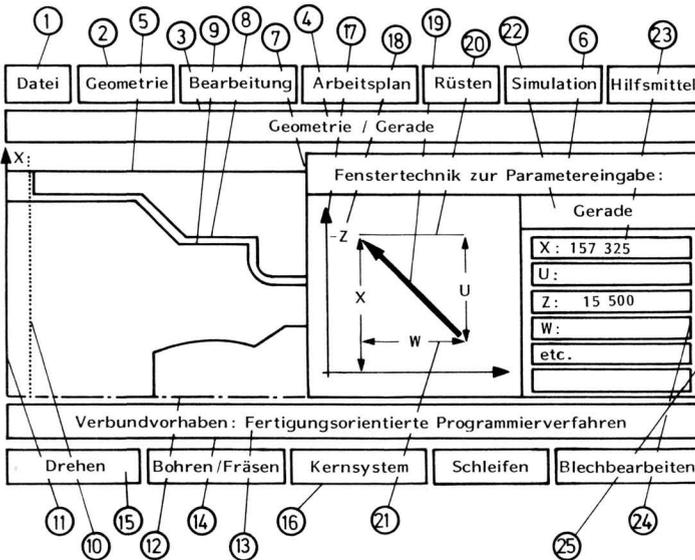


Bild 5: Elementarobjekte in einer CNC- Maske

- b) Bildung von Objektklassen: Alle Elementarobjekte, die z.B. bezüglich der Ausprägung des Gestaltungsparameters Farbe in den verschiedenen Dialogzuständen nicht weiter zu differenzieren sind, werden dynamisch zusammengefaßt.
- c) Zuordnung von Parameterausprägungen zu den einzelnen Objektklassen

Diese Strukturierung der Information ist in der Dialogsoftware voll nachvollzogen worden; sie erlaubt die freie Umstellung aller Parameterausprägungen im Dialog während der Entwicklungs- und Evaluationsphase.

In dem implementierten System wurde eine Klassenbildung entsprechend dem Aktualitätsgrad der einzelnen Objekte gebildet. Zusammengefasst sind demnach Felder und Parameter für Benutzereingaben, Bildelemente zur unterstützenden Beschreibung möglicher Interaktionen sowie Hintergrundinformationen wie Hilfslinien, Rahmen etc.

## 5. Software- Werkzeuge

Schichtenarchitekturen für Benutzerschnittstellen bieten auch aus Sicht des Software- Engineerings klare Vorteile, die in /8/ diskutiert sind. Dort wird darauf hingewiesen, daß Software- Werkzeuge zur Realisierung von Benutzerschnittstellen deren Spezifikation und Implementation effizienter gestalten und die softwareergonomische Qualität des Produktes verbessern. Eine (nicht vollständige) Zusammenstellung gängiger Werkzeuge findet sich in /9/.

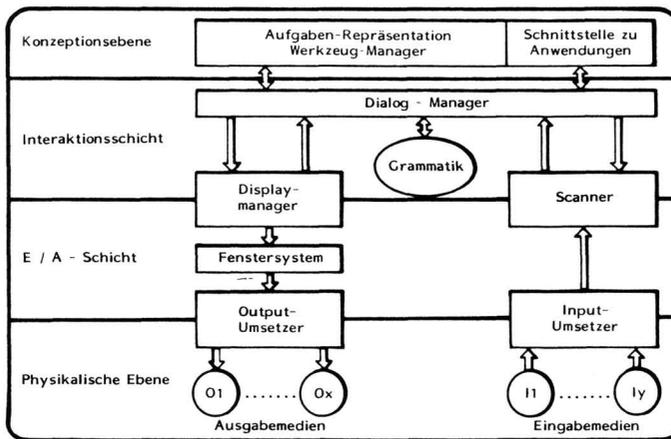


Bild 6: Einige Komponenten einer geschichteten Softwarearchitektur für Benutzerschnittstellen

In der folgenden Übersicht sind den verschiedenen Schichten jeweils entsprechende Implementationswerkzeuge zugeordnet:

- o Ein-/Ausgabeschicht:
- o Input/ Output- Handler
- o Fenstersysteme, Display- Manager, Piktogramm-Manager
- o Graphik-Pakete (z.B. GKS, Quick-Draw)

- o Dialogschicht:
- o Dialog-Manager
- o Menü- und Formular-Manager (Textmenü, Formular, Piktogramm- Menü)
- o Scanner bzw. Tokenizer (Zerlegung von Eingabesequenzen in Symboleinheiten)
- o Parser für die syntaktische Analyse von Token-Sequenzen
  
- o Werkzeug und Aufgabenschicht:
- o Interpreter oder Compiler für Zustands-Transitionsnetze (Präsent. des Aufgabenraumes).
- o Environment Manager, Werkzeug (Tool) - Manager, Shells

Kommerziell verfügbare Produkte wie GEM /10/ oder die "Toolbox"-Entwicklungsumgebung für APPLE Lisa / Macintosh /11/ unterstützen vor allem den Bereich der E/A-Funktionalität und unteren Dialogfunktionalität im Rahmen eines relativ stark festgelegten konzeptuellen Modelles (objektorientierte, direktmanipulative Schnittstelle). Kommandosprachliche Interfaces oder generelles Dialogmanagement werden nicht unterstützt. Innerhalb der GEM- Entwicklungsumgebung ist ein komfortabler Resourceeditor für die Erstellung von Dialogboxen Menülischen, Formularen und Piktogrammen verfügbar. Auf Seiten der E/A-Schnittstelle ist durch die Bereitstellung oben genannter Werkzeuge eine relative Unabhängigkeit vom Ein-/Ausgabesystem gegeben. Beide Entwicklungsumgebungen wurden für die Realisierung der in Kap. 6 beschriebenen Systeme eingesetzt.

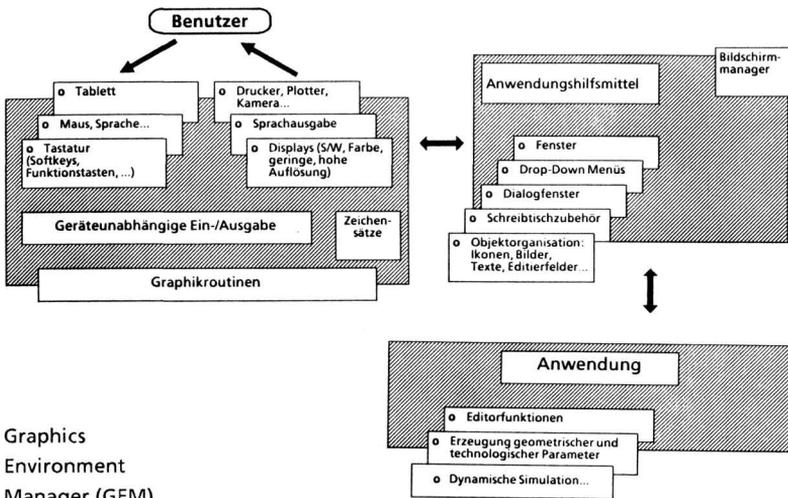


Bild 7: Graphics Environment Manager (GEM)

Sogenannte "User Interface Management Systems" (UIMS), die auch die höhere Dialogfunktionalität abdecken, befinden sich derzeit noch in einem experimentellen Stadium. Beispiele dafür sind die Systeme XS-2 /12/ und COUSIN /13/, /14/. Solche Systeme beruhen in der Regel auf einem oder mehreren Repräsentationsmechanismen für Dialogsyntax und -semantik, z.B. "Erweiterten, Attribuierten Grammatiken" (EAG's) oder "Augmented Transition Networks" (ATN's).

---

Überlegungen zu Softwarewerkzeugen orientieren sich an Arbeiten von Dr. Ulrich Hoppe, IAO

Solange hochstehenden Systeme noch nicht zur Verfügung stehen, kann die Entwicklung eigener Werkzeuge durch die Verwendung von "Meta-Tools" wie z.B. Compiler-Compilern oder Programmierumgebungen aus dem Bereich der "Künstlichen Intelligenz" (Expert System Shells) unterstützt werden. Im vorgestellten Vorhaben wurden entsprechende Metawerkzeuge geschaffen:

- o Ein Compiler für Transitionsnetze zur Aufgaben- und Dialogrepräsentation.
- o Eine Shell für ein regelbasiertes Expertensystem.

Darüber hinaus wurden die Mehrzahl der vorher diskutierten Komponenten im Wesentlichen mit Ausnahme eines komfortablen Fenstersystems implementiert. Das folgende Bild zeigt einen Compiler, der mit Hilfe von Compiler- Compiler Techniken generiert wurde und die Übersetzung der formalen Spezifikation zur Aufgaben- und Dialogrepräsentation (Transitionsnetze) in ein ausführbares C- Programm vornimmt.

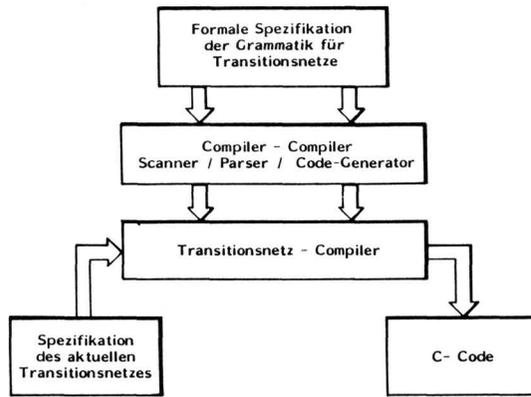


Bild 8: Einsatz von Compiler- Compiler- Techniken

Das folgende Bild zeigt den Aufbau der Shell für ein regelbasiertes Diagnosesystem, bei dem der Wissensrepräsentationsmechanismus ähnlich wie bei sogenannten "Produktionssystemen" realisiert wurde. Für die Shell wurde ein Interpreter implementiert, der das Testen des Systems auf dem Entwicklungssystem zuläßt. Ein Compiler erzeugt das lauffähige Programm in der Sprache C für die Implementation auf dem Zielsystem der Maschinensteuerung.

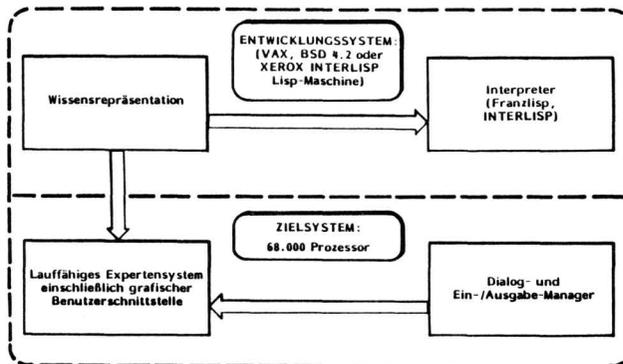


Bild 9: Shell für ein Expertensystem zur Fehlerdiagnose an CNC- Maschinen

## 6. Weit fortgeschrittene Systeme aus dem Bereich des Computer Integrated Manufacturing (CIM)

Ausgehend von dem zentralen Bereich der CNC- Programmierung wurden Arbeiten an weiteren Systemen aus dem Produktionbereich ausgeführt. Lösungen sind zu den folgenden Themenbereichen erarbeitet worden:

- o Die nächste Generation graphisch interaktive Dialogprozessoren für das Drehen und weitere Bearbeitungsverfahren wie Fräsen, Stanzen/Nibbeln, und Schleifen.
- o Leitwarte zur Fernbedienung unterschiedlicher Maschinensteuerungen.
- o Produktions-Planungs-und Steuerungssystem (PPS).
- o Expertensystem für die Fehlerdiagnose an Werkzeugmaschinen

Die folgenden beiden Bilder zeigen eine weit fortgeschrittene Lösung für Programmierumgebungen mit einer direkt manipulativen Benutzerschnittstelle, die unter Einsatz der Macintosh Toolbox implementiert wurde. Dabei wurde besonders auf die Konsistenz der Umgebung für unterschiedliche Bearbeitungsverfahren (generische Objekte und Funktionen) Wert gelegt.

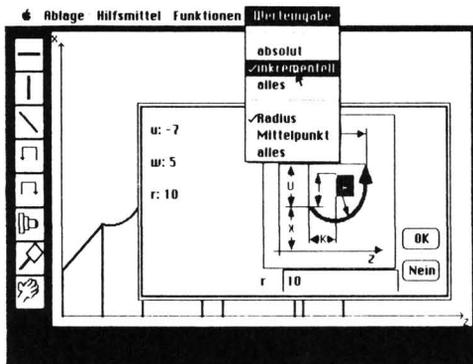


Bild 10: Programmierplatz für die Drehbearbeitung

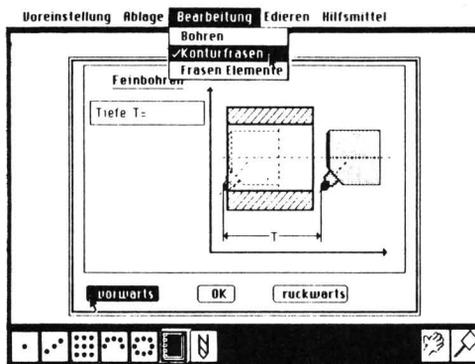


Bild 11: Programmierplatz für die Fräsbearbeitung

In einem weiteren Vorhaben wurde ein Produktions-Planungs- und Steuerungssystem entwickelt. Ein wesentliches Ziel dabei war, einen PC als "lokales intelligentes Terminal" zu nutzen. Der PC ist hierbei über eine direkte Rechnerkopplung mit einem Hostrechner und zentralem Datenbestand verbunden. Durch diese Systemkonfiguration lassen sich grundsätzlich neue Interaktionsformen für die Arbeit an PPS-Systemen finden. Als Dialogtechnik konnte hier das Prinzip der Direkten Manipulation verwendet werden. Dem Benutzer wird dabei in visualisierter Form eine Informationsumgebung angeboten, die z.B. einer Werkstatt/Büroumgebung entspricht.

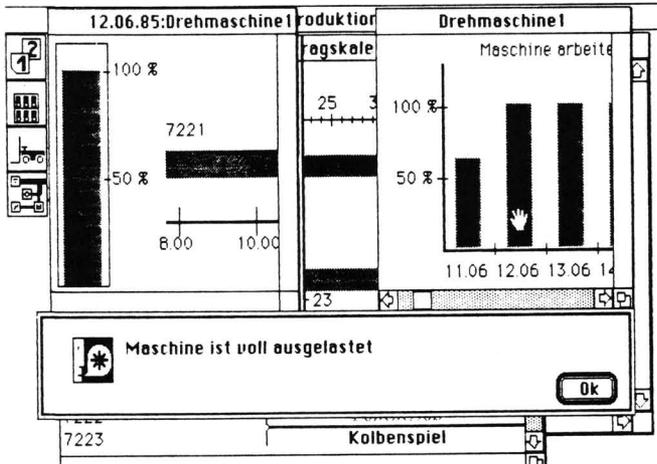


Bild 12: Produktions-Planungs- und Steuerungssystem

Forschungsarbeiten, die von ihrer praktischen Bedeutung für den Maschinenbau erst in den nächsten fünf Jahren voll zum Tragen kommen, liegen im Bereich von Expertensystemen. Bild 13 zeigt einen Teil der Benutzerschnittstelle eines solchen Expertensystems zur Fehlerdiagnose an CNC-Maschinen, bei dem der Benutzer im Störfalle durch Vorgabe potentieller Fehlerbereiche die Diagnose durchführen kann. Die Auswahl erfolgt entweder über Positionierung eines Zeigers in der Graphik oder über die Softkeyleiste.

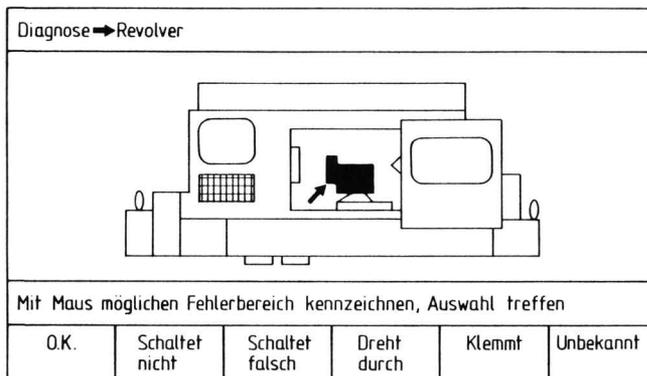


Bild 13: Expertensystem zur Fehlerdiagnose an CNC-Maschinen

Die momentane Lösung wurde in OPS 5 und Franzlisp /15/ implementiert und mittels eines eigenentwickelten Compilers in C-Code für das Zielsystem umgesetzt. Für die Maschinendiagnose sind die Erfassung von bis zu 40.000 Fehlerzuständen vorgesehen.

Alle vorher geschilderten Arbeiten zielen in die Richtung, Interaktion mit dem System in einer Weise zu realisieren, die den Anforderungen aus Aufgabe und Benutzergruppe Rechnung trägt und den Einlernaufwand minimiert. So werden arbeitsorganisatorische Lösungen ermöglicht, die Planungs- und Programmieraufgaben maschinennah bzw. im Werkstattbereich erlauben. Damit wird ein Beitrag geleistet, Dequalifizierungstendenzen im Werkstattbereich beim Einsatz neuer Informations- und Kommunikationstechnologien zumindest nicht zu erzwingen.

#### Literatur:

- /1/ Bullinger, H.-J., Fähnrich, K.-P., Raether, C.: Task and User Adequate Design of Man-Computer Interfaces in Produktion, In: Shackel, B. (Hrsg.): Proc. 1st Conf. Human- Computer- Interaction, INTERACT 84, London, 1984.
- /2/ K.-P. Fähnrich, P. Kern: Benutzer- und aufgabengerechte Programmierschnittstellen an CNC-Maschinen. In: FHG-Berichte, 3/4, 1983.
- /3/ Fähnrich, K.-P.; Ziegler, J.: Workstations Using Direct Manipulation as Dialog Principle - Aspects of Design, Application and Evaluation. In: Shackel, B. (Hrsg.): Proc. 1st Conf. Human-Computer- Interaction, INTERACT 84, London, 1984.
- /4/ Fähnrich, K.-P.; Ziegler, J.: Direkte Manipulation als Interaktionsform an Arbeitsplatzrechnern: In: Proc. der Software-Ergonomie 85, Tagung Mensch-Maschine-Kommunikation, Teubner, Stuttgart, 1985.
- /5/ Dzida, W.: Das IFIP-Modell für Benutzerschnittstellen. In: Office Management 31, (1983). Sonderheft: "Mensch-Maschine-Kommunikation".
- /6/ Ziegler, J.; Fähnrich, K.-P.; Raether, C.: Textsysteme und ihre Benutzerschnittstelle. In: Proc. Offene Multifunktionale Büroarbeitsplätze und Bildschirmtext, TU Berlin 1984.
- /7/ Dzida, W.: Ergonomische Normen für die Dialoggestaltung: In Proceedings der Software-Ergonomie 85, Tagung Mensch-Maschine-Kommunikation, Teubner, Stuttgart, 1985.
- /8/ Fähnrich, K.-P.; Kärcher, M. : Software-Architekturen für Mensch-Computer-Schnittstellen - dargestellt am Beispiel eines multilingualen Textsystems. In: Proc. der Software-Ergonomie 85, Tagung Mensch-Maschine-Kommunikation, Teubner, Stuttgart, 1985.
- /9/ Unterlagen aus: Shakel, B. (Hrsg.): Proc. NATO Workshop on Research Needs in Human-Computer- Interaction, Loughborough, 1985.
- /10/ Miller, M.J.: Digital Research`s Graphics Environment Manager (GEM). In: Popular Computing 14, (1985), 5, S. 19-20.
- /11/ Williams, G.: The LISA Computer System. In: BYTE, 8, (1983), 2, S. 33 ff.
- /12/ Stelovsky, J.: XS-2: The User Interface of an Interactive System; Diss. ETH Zürich, 1984
- /13/ Hayes, P.J.; Szekeley, P.A.; Lerner, R.A.: Design Alternatives for the User Interface Management Systems based on Experience with Cousin. In: Proc. CHI `85, Human Factors in Computing Systems, April 14-18, San Francisco 1985.
- /14/ Hayes, P.J.; Szekeley, P.A.: Graceful Interaction through the Cousin Command Interface. In: Int. Journal of Man- Machine Studies 1b,3, (Sept. 1983), S. 285- 305.
- /15/ AI Language Programs: Expert Systems Under VAX/VMS Computers. In: Design, 23, (1984), 14, S. 239.