

InterCon - Ein interaktives Werkzeug zur Konfiguration graphischer CAD-Benutzungsoberflächen

Willi Schwier, Geerd Philipsen, Ralf Garbade,
Carsten Hoppe, Roland Kubica, Andreas Röpneck
Universität Bremen

Zusammenfassung

Die Konfigurierbarkeit der Benutzungsoberfläche von CAD-Systemen unabhängig von den Entwicklungszyklen der Applikation wird als zentrale Voraussetzung für die Anpassung der Benutzungsoberfläche an die jeweilige Arbeitssituation angesehen.

Die Konfiguration wird als kooperativer Arbeitsprozeß von Entwicklern, Anwendern und Software-Ergonomen betrachtet, den es durch ein geeignetes Werkzeug zu unterstützen gilt. Die Konzeption eines solchen Werkzeuges - **Interface Configurator** - wird vorgestellt. InterCon selbst ist Bestandteil der Werkzeugschnittstelle des CAD-Systems, dessen Ein-/Ausgabeelemente und Dialogstruktur das Werkzeug abbildet und verwaltet und für die modifizierende und generierende Funktionen bereitgestellt werden. Ergonomische Gestaltungslösungen auf der Elementebene und als aufgabenorientierte Schnittstellen-Konzepte sind in Form von Defaults oder Auswahllisten implementiert.

1 Einleitung

Neben der Erfüllung grundlegender Forderungen nach schädigungslosen, ausführbaren, erträglichen und beeinträchtigungsfreien Arbeitsbedingungen sollte menschliche Arbeit aufgabenangemessen, persönlichkeitsförderlich und sozial sein (Luczak, Volpert, Raeithel & Schwier [8]). Sie sollte ganzheitliche und vielfältige Anforderungen aufweisen und Chancen zum Kompetenzerwerb und zur Qualifizierung in Bereichen bieten, die in Zusammenhang mit dem eigentlichen Arbeitsinhalt stehen. Ganzheitliche, vielfältige und qualifizierende Arbeit bildet "vollständige" Tätigkeiten aus (Hacker [5]), die als Ziel arbeitsgestaltender Maßnahmen auch die arbeitswissenschaftliche Gestaltung von Mensch-Rechner-Systemen leiten sollten. Für das Gestaltungsobjekt Benutzungsoberfläche, als ein wesentlicher Ausschnitt des durch das Mensch-Rechner-System beschriebenen Gestaltungsraumes, werden diese auf allgemeine Arbeitssysteme zielenden Gestaltungsanforderungen operationalisiert in Form von Kriterien und Empfehlungen (DIN 66234, ISO 9241). CAD-Benutzungsoberflächen weisen insbesondere hinsichtlich der Kriterien Aufgabenan-

gemessenheit und Steuerbarkeit bzw. "Suitability for Individualization" Gestaltungsdefizite auf (siehe Dzida [2]). Vielfach erschöpft sich die CAD-Arbeit für den Benutzer in repetitiver Systembedienung, was besonders bei wenig anspruchsvollen Tätigkeiten zu einer monotonen Arbeitssituation führen kann (Hacker [5]). Deshalb müssen Werkzeuge zur Verfügung gestellt werden, die es erlauben, die Schnittstelle so zu modifizieren, daß repetitive Abläufe unterbleiben und höherqualitativer Arbeit im Interaktionsprozeß mehr Raum eingeräumt werden kann.

Leider sieht die Konzeption der wenigsten existenten Systeme ein solches Werkzeug zur Anpassung der Benutzungsoberfläche vor, wie es schon in einigen Standardschnittstellen angetroffen wird. Ein solches Werkzeug sollte eine auf mehrere Zielgruppen ausgerichtete aufgabenangepaßte Gestaltung der Schnittstelle ermöglichen. Es sollte attraktiv im Design sein und zu seinem Gebrauch animieren. Im Dialog sollte das Werkzeug konsistent sein mit der Applikation, deren Teil es ist und effiziente Dialogformen unterstützen. Als ein solches Werkzeug konzipiert worden ist InterCon (**I**nter**f**ace **C**onfigurator), welcher im folgenden in seiner Funktionalität und weiter in Anlehnung an das IFIP-Benutzerschnittstellenmodell (Dzida [1], [2]) in seinen Ein-/Ausgabe-, Dialog-, Werkzeug- und Organisationsaspekten beschrieben wird. InterCon wird als Element der Werkzeugschnittstelle eines CAD-Systems in dessen Benutzungsoberfläche integriert. Ein erster Prototyp wird zur Zeit im Rahmen des vom BMFT geförderten Forschungsprojektes "Gestaltung einer Benutzerschnittstelle mit Werkzeugcharakter am Beispiel des CAD-Systems KONSYS", Förderkennzeichen 01HK530/4, realisiert. Das Projekt wird Ende 1993 abgeschlossen.

2 Gestaltungskonzept

In Fortsetzung der auf der Software-Ergonomie'91 vorgetragenen Überlegungen von Paetau [11] sowie Greutmann [3], [4] sieht das Gestaltungskonzept einen mehrschichtigen Aufbau vor, der verschiedene mögliche Anwenderkreise des Schnittstellenkonfigurators berücksichtigt. Anwender im Sinne dieses Gestaltungskonzeptes sind Personen, die mit dem Schnittstellenkonfigurator arbeiten. Dies können, je nach den organisatorischen Gegebenheiten, verschiedene Personengruppen sein. Sie können extern in Softwarehäusern, die applikations- und branchenspezifische Lösungen anbieten, in überbetrieblichen Beratungsinstanzen oder in internen Betriebs-, Gruppen- oder Arbeitsplatzstrukturen angesiedelt sein. Im letzten Fall würde dies Anpassung durch den "Benutzer" bedeuten, wobei der Begriff des Benutzers, hier ähnlich wie in den Gestaltungsempfehlungen der Gesellschaft für Informatik [2] formuliert, den Personenkreis kennzeichnet, der die Applikation im Rahmen seiner tatsächlichen Aufgabebearbeitung einsetzt.

Eine Anpassung der Schnittstelle durch den Benutzer ist im Rahmen dieses Gestaltungskonzeptes vorgesehen. In der Anpassung der Schnittstelle liegen umfassende Möglichkeiten zum Kompetenzerwerb und zur Qualifizierung (vgl. Ulich [14]). Die Auseinandersetzung mit den im System liegenden Gestaltungsmöglichkeiten wird zu mehr Transparenz des Systems, einer umfassenderen Beherrschung der Systemfunktionalität und zu einem subjektiven Effizienzepfinden führen, welches als grundlegende Vorbedingung effizienter, kompetenzerweiternder und motivierender Arbeit gilt (White [16]).

Auch von betriebsorganisatorischer Seite liegen die Vorteile einer benutzerseitigen Schnittstellenkonfiguration auf der Hand. Schon allein aufgrund der institutionellen, zeitlichen und räumlichen Trennung von Entwicklung und Anwendung ist der CAD-Systemanbieter nicht in der Lage, auf aktuelle Veränderungen in der Produktpalette des Unternehmens oder des Benutzerkreises zu reagieren. Dieses Problem ist zwar über die Delegation von Konfigurationsaufgaben an einen innerbetrieblichen Systembetreuer zu lösen, in einem anderen Punkt hat die Systemkonfiguration durch den Benutzer aber einen entscheidenden Vorteil: Nur die das CAD-System letztlich benutzende Person verfügt über das entscheidende aufgabenspezifische Fachwissen, was es ihr ermöglicht, Auslegungsdefizite in Bezug auf die konkrete Arbeitsaufgabe zu identifizieren und Verbesserungen zu konzipieren. In Hinblick auf seine Arbeit ist derjenige Experte, der die Arbeit ausübt. Es liegt nahe, gerade die Personen, die von Systemdefiziten am meisten betroffen sind, an deren Behebung und darüber hinaus an der Optimierung der Systeme in dem Sinne zu beteiligen, daß eine Anpassung an die arbeitsplatzspezifischen Gegebenheiten vorgenommen wird.

Des weiteren stehen die Betriebe vor der Notwendigkeit, CAD-Personal zu schulen bzw. schulen zu lassen. Wesentliche Anteile der Schulung werden jedoch in den eigentlichen Tätigkeitsprozeß selbst verlagert (Müller, Cords & Peters [9]) und erfolgen mittels Exploration des Systems durch den Benutzer mit den bekannten Problemen und Nachteilen, die mit der Ein-/Ausgabekomponente als "Fenster" zum System verbunden sind. Ein Konfigurationswerkzeug mit seinen die Struktur und Funktionalität des Systems abbildenden Elementen sollte den Qualifizierungsprozeß wesentlich unterstützen können und die Bildung von Inkompatibilitäten zwischen mentalen und rechnerinternen Modell (siehe Hartmann & Eberleh [6]) vermeiden helfen.

3 Funktionalität

Die Funktionalität beliebiger Dialogsysteme kann nach Rohr ([12], S. 40) unterteilt werden in Existenz- (Was?), Platz- (Wo?), Eigenschafts- (Wie?), Zeit- (Wann?), Ursachen- (Warum?) und Zielfunktionen (Wozu?). In ähnlicher Weise muß die

Funktionalität eines Werkzeugs zur Modifikation solcher Systeme diese Bereiche abdecken.

InterCon ist in erster Linie als Konfigurator bestehender Benutzungsoberflächen ausgelegt. Dementsprechend stehen modifikatorische Operationen (Eigenschaftsfunktionen), die die Ein-/Ausgabe-Schnittstelle sowie die Dialogschnittstelle des Systems verändern, im Vordergrund der Funktionalität dieses Werkzeugs. Unter Zeitfunktionen werden alle Funktionen definiert, die den zeitlichen Ablauf der Interaktion beeinflussen. Ursachenfunktionen dienen dazu, Ursache-Effekt-Zusammenhänge darzustellen und gegebenenfalls zu verändern. Ursachenfunktionen werden implizit über räumliche Gruppierungen verdeutlicht (z.B. geben Menükaskaden Auskunft über Ursache-Effekt-Zusammenhänge innerhalb eines Menüs), zum anderen werden damit explizite Informations-, Protokollierungs- und Dokumentationsprozeduren bezeichnet. Als eigentliche Zielfunktion des Schnittstellenkonfigurators wird die Erstellung und Auswahl einer aufgabenangepaßten und auf den Benutzerkreis zugeschnittenen Schnittstellenkonfiguration aufgefaßt.

Eigenschaftsfunktionen (modifizierende Funktionen):

Alle Ein-/Ausgabeobjekte mit den folgenden Eigenschaften sind auch in diesen veränderbar: Position, Anordnung (Abstand zu Nachbarelementen), Größe (Höhe/Breite), Farbe, Schrift (Font), Titel, Dialogtext, Bitmaps, Hintergrundmuster, Markierung.

Existenzfunktionen (generative und destruktive Operationen):

Obwohl der Konfigurator nicht notwendig auf Existenzfunktionen aufsetzt, sondern primär modifikatorische Funktionen besitzt, sind auch Existenzfunktionen für die Fälle zu berücksichtigen, in denen das Angebot an bestehenden Schnittstellenobjekten nicht ausreicht. Im Sinne einer ergonomischen Lösung kann es nicht sein, eine beliebige Definierbarkeit von Objekten zuzulassen. Indem neue Objekte geschaffen werden, andere Objekte gelöscht und Funktionen neu zugeordnet werden, greifen Existenzfunktionen tief in die Struktur der Benutzungsoberfläche ein.

Deshalb werden ergonomische Festlegungen in der Weise getroffen, daß nur bestimmte Kombinationen von Objekten erzeugbar sind. Das grundsätzliche Layout der Objekte ist z.Z. noch festgelegt durch die von InterViews (Linton, Vlissides & Calder [7]) bereitgestellten Elemente.

Existenzfunktionen sind Generieren, Löschen, Kopieren und Speichern von Objekten sowie das Zuordnen von Applikationsfunktionen zu Objekten.

Generell können mit dem Schnittstellenkonfigurator keine neuen Applikationsfunktionen und Parameter generiert werden, sondern es können die Applikationsfunktio-

nen lediglich (neuen oder bestehenden) Ein-/Ausgabeobjekten zugeordnet werden. Die Zuordnung von Ein-/Ausgabeobjekten zu Makros wird über eine definierte Schnittstelle zu einem Makrorecorder/-editor gewährleistet.

Die Struktur der Ein-/Ausgabeobjekte folgt weitgehend einer Parent-Child-Hierarchie. Als Basisobjekt (root) der erzeugten Ein-/Ausgabeobjekte wird das Applikationsfenster betrachtet, in dem der Volumenmodellierer läuft. Das Applikationsfenster selbst mit seinen direkt zugehörigen Elementen (Titelzeile, Rahmen, etc.) wird nicht durch den Schnittstellenkonfigurator, sondern durch das CAD-System selbst bzw. den zugrundeliegenden Window-Manager verwaltet. Den Rang unterhalb dieser obersten Hierarchieebene nehmen Felder (Sheets) ein. Unterhalb dieser Ebene liegen die diversen Bedienelemente ("Controls"), die wiederum durch spezifische Subobjekte (graphische Symbole, Titel, Marken, etc.) gekennzeichnet sind. Ein Durchbrechen dieser Objekthierarchie ist teilweise erlaubt. Felder können weitere Felder enthalten (z.B. in Menüs) oder an ein Control angebunden sein. Nicht erlaubt ist, Controls direkt auf ein Applikationsfenster zu legen, ohne ein sekundäres Fenster oder Feld als aufrufendes Objekt zu definieren.

Felder (Sheets):

Es werden vier Instanzen von Feldern eingerichtet: Dialogfelder, Propertyfelder, Controllfelder und Menüfelder. Dialogfelder, in einem anderen Kontext als "command box" (Sun [13]) bezeichnet, werden eingesetzt, um alle zum Auslösen einer Aktion notwendigen Spezifikationen vorzunehmen. Solche Dialogboxen sind vor allem bei komplexen Operationen notwendig und hilfreich, da sie alle notwendigen Parameter der Operation zusammenfassen. Dialogboxen können wie Propertyfelder alle Arten von Controls enthalten. Sie unterscheiden sich visuell von diesen dadurch, daß der sie abschließende Aktionsbutton den Namen der ausgelösten Applikationsfunktion enthält. Propertyfelder stellen im Gegensatz zu Dialogfeldern keine Parameter von Applikationsfunktionen, sondern Objektattribute (des gerade aktiven Objektes) dar. Diese Attribute können prinzipiell alle Ausprägungen auf den Variationsdimension eines Ein-/Ausgabeobjektes darstellen. In der Regel werden in Propertyfeldern Attribute wie Größe, Position, Farbe etc. gesetzt. Veränderte Attributwerte können durch die Aktivierung eines Aktionsbuttons ("Apply") gesetzt bzw. widerrufen oder abgebrochen werden ("Reset") werden. Diese Aktionsbuttons sind zwingend vorhanden und werden automatisch bei der Generierung eines Propertyfeldes eingefügt. Fakultativ können zusätzlich Aktionsbuttons zur Defaulteinstellung und zur Erweiterung der Einstellungen auf weitere Selektionen eingefügt werden. Andere Aktionsbuttons sind nicht zulässig. Propertyfelder besitzen eine Default-Position und können an dieser Stelle "verankert", d.h. permanent gesetzt werden. Propertyfelder erhalten wie Dialogfelder einen Titel.

Controllfelder dienen zur Anordnung mehrerer Bedienelemente gleichen Typs nach unterschiedlichen Strukturierungsprinzipien. Menüfelder enthalten lediglich Menüitems und/oder Aktionsbuttons, aber keine weiteren Controls. Control- und Menüfelder enthalten eine Default-Position und können intransient gesetzt werden.

Controls/Bedienelemente:

Controls sind letztlich die Objekte, mit deren Hilfe Aktionen ausgelöst, Parameter eingestellt und Informationen abgerufen werden. An Controls sind generierbar: Aktionsbuttons zum Auslösen von Aktionen, Menüitems, Schalter ("Toggles") für binäre Wahl, Radiobuttons (single choice), Multi-Choice-Buttons, Scrolllisten zur Auswahl mehrerer diskreter Ausprägungen, Schieberegler ("Slider") zur ungefähren Einstellung kontinuierlicher Werte, Skalen zum Abruf numerischer Werte, Numerikfelder zur genauen Einstellung diskreter numerischer Werte, Textfelder zur Eingabe von kurzem Text, Textregion, Bitmaps, Titel, Markierungen.

Zeitfunktionen (Dialog):

Zeitfunktionen werden einerseits implizit durch die Spezifikation des Dialogablaufs der Ein-/Ausgabe-Objekte bestimmt, zum anderen durch die Regeln, die invariant für die Dialogschnittstelle der Applikation (und damit auch für ihre Werkzeugschnittstelle) gelten.

Ursachenfunktionen:

Ursachenfunktionen werden ebenfalls über den Dialogablauf der Ein-/Ausgabeobjekte, zum anderen über die "Werkzeugschnittstelle" des Werkzeugs "Konfigurator" implementiert. Informations-, Kommunikations- und Protokollierungsregeln werden dabei festgelegt.

Zielfunktionen (Einsatzzweck):

Als eigentliche Zielfunktion des Schnittstellenkonfigurators ist die Modifikation der existenten Benutzungsoberfläche einer Applikation zu betrachten. Zu dieser Zielfunktionalität gehört:

- die Auswahl einer Schnittstellenkonfiguration,
- das Rücksetzen der Konfiguration auf einen Standard,
- das Abspeichern der geänderten Konfiguration sowie,
- das Setzen eines Konfigurations-Standards.

Bis auf das Setzen eines Konfigurations-Standards sollten dem Benutzer alle Zielfunktionen zugänglich gemacht werden.

4 Die Ein-/Ausgabe-Schnittstelle

Die Ein-/Ausgabe-Ebene von InterCon setzt sich aus den gleichen Objekten zusammen, die InterCon selbst benutzt, im wesentlichen also Felder und Controls. Eine Besonderheit von InterCon ist, daß das Werkzeug optisch als Fenster innerhalb des Grafikfensters der Basisapplikation erscheint, wobei die Benutzungsoberfläche der CAD-Basisapplikation visuell in der Form erhalten bleibt, in der sie sich während der eigentlichen Systembedienung darstellt. Die Ein-/Ausgabeobjekte der Applikation werden durch den Schnittstellenkonfigurator in der gleichen Größe und an der gleichen Position wiedergegeben wie in der realen Benutzungsoberfläche. Dieses Layout hat den Vorteil einer unmittelbar kompatiblen und transparenten Abbildung der für die Konfiguration relevanten Objekte. Zumindest die visuelle Rückmeldung über die Güte einer Veränderung erfolgt unmittelbar und ermöglicht Vorher-Nachher-Vergleiche in annähernd realer Umgebung. Abb. 1 verdeutlicht diese Auslegung. Das zunächst ungewöhnliche "reziproke" Verhältnis zwischen Werkzeug und zu konfigurierendem Objekt, was vielleicht auch beim Benutzer anfangs zu Irritationen führen kann, wird in Anbetracht der unmittelbaren und realistischen Abbildung des eigentlich interessierenden Gestaltungsobjektes, der Benutzungsoberfläche der Applikation, in Kauf genommen.

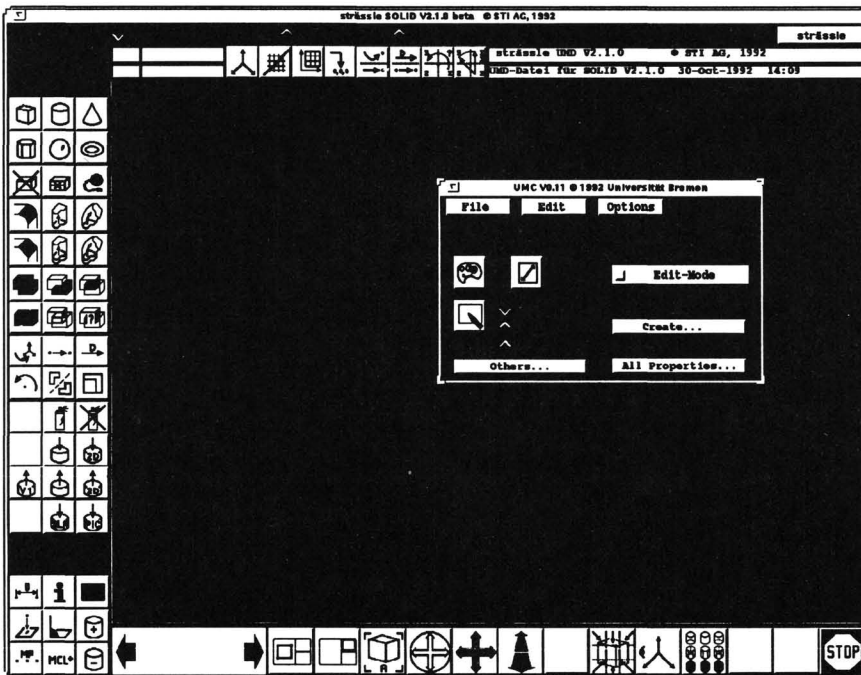


Abb.1: Ein-/Ausgabeschnittstelle von InterCon

Der Bedienbereich des Schnittstellenkonfigurators gliedert sich in Anlehnung an den OSF/Motif Standard (Open Software Foundation [10]) in eine horizontale Menüzeile mit den Items "File", "Edit" und "Options" und in ein Dialogfeld, in dem die verändernden und generierenden Funktionen des Schnittstellenkonfigurators angeboten werden.

5 Die Dialogkomponente

Grundsätzlich soll die Dialogstruktur einer Objekt-Aktions-Syntax (objektorientierter Dialog) folgen. Die Objekt-Aktions-Syntax setzt die Selektion eines zu konfigurierenden Ein-/Ausgabeobjektes voraus, bevor Operationen auf diesem Objekt ausgeführt werden können. Die Objekt-Aktions-Syntax bietet den großen Vorteil, nur die auf das selektierte Objekt anwendbaren Funktionen aktiv zu halten, die übrigen Funktionen werden auf "nicht verfügbar" gesetzt. Zudem werden durch die Objekt-Aktionssyntax die spezifischen Attribute des gerade selektierten Objektes

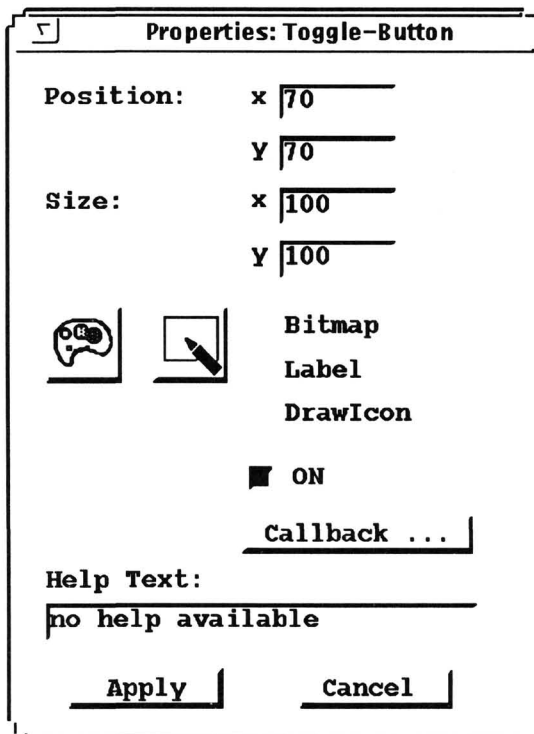


Abb. 2: Property-Feld für den Toggle-Button

darstellbar, was die Anwendung von Propertyfeldern erst ermöglicht. Das gleichzeitige Aktivieren mehrerer Objekte, welches in einem operationsorientierten Dialog über Abfragen erfolgt, kann in der objektorientierten Syntax durch Modifikatortasten oder Aktionsbuttons, durch die die Selektion ausgedehnt werden kann ("Apply to new selection", vgl. Sun [13]), realisiert werden.

Zwangsläufig kann eine objektorientierte Syntax für die generischen Funktionen nicht durchgehalten werden (da es beim Auslösen der Funktion noch keine aktivierbaren Objekte gibt). Dennoch kann auch hier der Vorteil des Propertyfensters ausgenutzt werden, in dem alle relevanten Objekteigenschaften zusammengefaßt präsentiert werden (siehe

Abb. 2). Die Einstellungen in diesem Propertyfenster können direkt über die entsprechenden Controls vorgenommen werden. Beim Aktivieren der generischen Funktion zeigt das Propertyfenster sinnvolle Defaultwerte, die einfach übernommen werden können. Das Propertyfenster wird erst dann entfernt, wenn die generische Aktion explizit beendet wurde.

6 Die Werkzeugkomponente

Die Werkzeugkomponente sieht definierte Schnittstellen zum Makrogenerator, zu einem Bitmap-Editor sowie zu Kommunikations-, Informations-, Dokumentations- und Protokollroutinen vor. Eine Schnittstelle zu einem Archiv und zum Datenaustausch mit anderen Applikationen wird vorgesehen.

Mittels der Schnittstelle zum Makrogenerator können Makros durch die Aufzeichnung einer Sequenz von Arbeitsschritten erzeugt werden (Recorder-Funktion), ohne daß dafür Programmier- oder Betriebssystemkenntnisse erforderlich sind (siehe Dzida [2], S. 43). Daneben soll mit dem Makrogenerator auch "klassisches" Editieren der Makros möglich sein. Über die Schnittstelle zum Makroeditor können Makros Aktionsbuttons zugewiesen werden. Mittels der Schnittstelle zum Bitmap-Editor lassen sich die Bitmaps der Buttons, sowie gegebenenfalls Hintergrundmuster von Feldern generieren und verändern.

Auf diese Weise sollen externe Informations- und Dokumentationsroutinen sowohl in die Applikation eingebunden (Zugriff auf Literaturdatenbanken, Normteilkataloge, Tabellenwerke etc.) als auch genutzt werden, um Veränderungen an der Benutzungsoberfläche zu dokumentieren und abfragen zu können. Die laufende Protokollierung der Konfigurationsarbeit erscheint zunächst nicht als notwendig, ist bei Bedarf allerdings realisierbar. Weiterhin wird eine Schnittstelle zu "electronic-mail"-Applikationen als notwendig erachtet, über die Vorschläge, Informationen und Ergebnisse von Konfigurationsänderungen ausgetauscht werden können.

7 Organisationskomponente

Die Anpassung der Benutzungsoberfläche von Software-Produkten für unterschiedliche Arbeitssituationen wurde von Paetau [11] als Kooperationsproblem thematisiert und die Ausweitung der persönlichkeitsbezogenen Anpassungsdiskussion "zu einer aufgabenbezogenen Modifizierbarkeit" begründet. Für die Auslegung eines Menü-Konfigurationswerkzeuges bedarf es demnach einer Gestaltungslösung, die zumindest die Einbindung des Konfigurationsprozesses in den jeweiligen betriebsorganisa-

torischen Kontext erleichtert. Ein möglicher Ansatz wäre die Festlegung von Zugriffsrechten auf die einzelnen Objekte oder Objektgruppen der Benutzungsoberfläche und deren Eigenschaften in Abhängigkeit vom Anwenderkreis. Aufgabenbezogen sind für CAD-Systeme die folgenden Anwenderkreise denkbar:

- Applikationsentwickler, die für Branchen oder auch branchenübergreifend Benutzungsoberflächen entwerfen.
- Systembetreuer in den Betrieben, die die Benutzungsoberflächen den firmenspezifischen Gegebenheiten anpassen. Auf dieser Ebene stellt sich die Frage der Zugriffsberechtigung nur insofern, als die Software-Häuser an solchen Gestaltungsmerkmalen festhalten wollen, die eine firmenbezogene Identifizierung des Produktes leisten.
- Arbeitsgruppen in den Betrieben, deren Mitglieder CAD-Systeme gemeinsam benutzen oder in kooperative Arbeitsprozesse eingebunden sind. Der Arbeitsgruppe sind weitgehende Konfigurationsrechte einzuräumen, da auf dieser Ebene aufgabenbezogene Anpassung der Benutzungsoberfläche am ehesten leistbar ist.
- Der Endbenutzer mit seinen persönlichen Vorstellungen hinsichtlich der Individualisierung der Benutzungsoberfläche. Die Konfigurationsmöglichkeiten wären nur für solche Eigenschaften bereitzustellen, die den grundsätzlichen Gestaltungskonsens der Arbeitsgruppe nicht in Frage stellen.

8 Ausblick

Mit InterCon sind die Anwender in der Lage, die Benutzungsoberfläche weitgehend nach den eigenen Vorstellungen zu gestalten. Abgesehen von einigen in das Werkzeug implementierten Restriktionen, wie z.B. die Zuordnung von manipulativen Applikationsfunktionen ausschließlich zu Aktionsbuttons, ist er frei in der Wahl seiner eigenen Gestaltungsvarianten. So kann er die ergonomische Gestaltung der "Default-Applikationsschnittstelle" durch eigenes Zutun verschlechtern, in dem er beispielsweise schlecht zu differenzierende Farbkombinationen wählt, Gruppierungsprinzipien mißachtet, oder in der Zuordnung von Applikationsfunktion zu Objekten ergonomisch suboptimale Entscheidungen trifft. Es stellt sich also das Problem, daß ein die Kompetenz und Handlungsfreiheit erweiterndes Werkzeug, notwendige Voraussetzungen humaner Arbeit beeinträchtigen kann. Die Lösung des Problems kann auf zwei Arten angegangen werden: (1) Über die genannte Differenzierung der Zugriffsrechte für verschiedene Anwenderkreise sowie (2) der Einführung software-ergonomischen Wissens in den Schnittstellenkonfigurator. Die letztgenannte Alternative bietet mehrere abgestufte sinnvolle Möglichkeiten, die Ergonomie der erstellten "neuen" Konfigurationen abzusichern. Zu denken wäre zunächst an ergonomisch günstige Default-Einstellungen, die beim Erzeugen neuer

Objekte übernommen werden können. In einer weiteren Stufe könnten mehrere ergonomisch begründbare Varianten in Form von "Paletten" oder Listen zu Verfügung gestellt werden. Eine regelbasierte Objektauswahl und Attributzuordnung, wie sie von Weisbecker [15] beschrieben wird, bietet weitere Steuerungsmöglichkeit für den Konfigurationsprozeß, bis hin zum automatischen Verhindern ergonomisch unsinniger Gestaltungslösungen. Einwände gegen die letztgenannten Realisationen sind besonders in Bezug auf den stark restriktiven Charakter dieser Eingriffe zu erheben, welche die durch den Schnittstellenkonfigurator gewonnenen Handlungsspielräume gefährden. Das Anbieten "ergonomisch geprüfter" Defaults, Optionen bzw. die regelbasierte Generierung von "Vorschlägen" sind demgegenüber nicht mit solchen Restriktionen verbunden. Sie bieten den Benutzern die Möglichkeit, sich ausdrücklich gegen einen Vorschlag zu entscheiden. Insofern unterstützen sie weitestgehend die Autonomie und Eigenverantwortlichkeit des Benutzers und sind damit weniger flexiblen Regeln vorzuziehen.

9 Literatur

- [1] Dzida, W. (1983). Das IFIP-Modell für Benutzerschnittstellen. Office Management, 31, Sonderheft, 6-8.
- [2] Dzida, W. (1988). Modellierung und Bewertung von Benutzerschnittstellen. Software Kurier, 1, 13-28. Gesellschaft für Informatik (1991). Gestaltungsempfehlungen für Benutzungsoberflächen von CAD-Systemen. Fachausschuß 4.2, Fachgruppe 4.2.1, Arbeitskreis AK 2.
- [3] Greutmann, Th. (1991). Die Konstruktion von Benutzungsschnittstellen: Ein neuer Ansatz für Systemarchitektur und Werkzeuge. In: D. Ackermann & E. Ulich (Ed.), Software-Ergonomie '91, pp. 252-261. Stuttgart: Teubner.
- [4] Greutmann, Th. (1992). HIDE and IDEA: Tools for User-Oriented Application Development. Informatik-Diss. ETH Zürich Nr.35, abstracted in Ergonomie & Informatik Nr.16, Juli 1992. Zürich: Verlag der Fachvereine.
- [5] Hacker, W. (1988). Zu Analyse-, Bewertungs- und Gestaltungsgrundlagen rechnergestützter Arbeit. In: E. Frieling, H. Klein (Ed.). Rechnerunterstützte Konstruktion: Bedingungen und Auswirkungen von CAD, pp. 40 - 48. Huber.
- [6] Hartmann, E.A. & Eberleh, E. (1991). Inkompatibilitäten zwischen mentalen und rechnerinternen Modellen im rechnerunterstützten Konstruktionsprozeß. In: D. Ackermann & E. Ulich (Ed.), Software-Ergonomie '91, pp. 301-310. Stuttgart: Teubner.
- [7] Linton, M.A., Vliissides, J.M. & Calder, P.R. (1989). Composing user interfaces with InterViews. IEEE Computer, 22(2), 8-22.
- [8] Luczak, H., Volpert, W., Raeithel, A. & Schwier, W. (1989). Arbeitswissenschaft - Kerndefinition, Gegenstandskatalog, Forschungsgebiete. Köln: TÜV Rheinland.
- [9] Müller, W., Cords, D. & Peters, W. (1990). Computer Aided Design und Ingenieurarbeit. Gestaltungsperspektiven und Rationalisierungsverständnis von Ingenieuren bei der Entwicklung, Einführung und Anwendung von CAD-Systemen. Forschungsbericht FNK 935, Universität Bremen.

- [10] Open Software Foundation (1990). OSF/Motif Style Guide. Englewood Cliffs, NJ: Prentice-Hall.
- [11] Paetau, M. (1991). Systemanpassung als Kooperationsproblem. In: D. Ackermann & E. Ulich (Ed.), Software-Ergonomie '91, pp. 281-290 Stuttgart: Teubner.
- [12] Rohr, G. (1988). Grundlagen menschlicher Informationsverarbeitung. In: H. Balzert, H.U. Hoppe, R. Oppermann, H. Peschke, G. Rohr & N.A.Streitz (Ed.), Einführung in die Software-Ergonomie , pp.27-48. Berlin: De Gruyter.
- [13] Sun Microsystems, Inc. (1990). Open Look Graphical User Interface Application Style Guidelines. Reading, MA: Addison-Wesley.
- [14] Ulich, E. (1991). Arbeitspsychologie. Stuttgart: Poeschel.
- [15] Weisbecker, A. (1992). Regelbasierte Generierung software-ergonomischer Benutzerschnittstellen aus Datenmodellen. Ergonomie & Informatik Nr.16, 19-27, Juli 1992.
- [16] White, R.W. (1959). Motivation reconsidered: The concept of competence. Psychological Review, 66, 297-333.

Dr.-Ing. Willi Schwier
Dr. Geerd Philipsen
Arbeitsgruppe KONSYS
Fachbereich Produktionstechnik
Universität Bremen
Badgasteiner Str. 1
2800 Bremen 33