

**MENÜGENERATOR: Ein Software-Entwicklungswerkzeug
zur interaktiven Generierung von Menü-Rahmenprogrammen**

Ludwig NASTANSKY
Wolfgang BERNARTZ

Zusammenfassung. Im nachfolgenden werden die Leistungsmerkmale, die Grundlagen der Benutzermaschine sowie die Eckdaten der Basismaschine des interaktiven Systems MENÜGENERATOR zur Generierung von menügesteuerten Rahmenprogrammen vorgestellt. Häufige Anwendung und vergleichsweise hohe Benutzerfreundlichkeit der Menütechnik für ein große Klasse von Dialog-Anwendungsprogrammen war eine der Motivationen für die Entwicklung von MENÜGENERATOR. Die programmtechnische Realisierung von Menüdarstellung und Ablaufsteuerung ist damit durch einen Programmgenerator zu einem Teil automatisierbar. Eine weitere Motivation ging von der Idee aus, in das Software-Werkzeug eine Menge von Utilities für die Gestaltung der Benutzeroberfläche zu integrieren. Dies erlaubt, den erfahrungsgemäß hohen Aufwand bei der Berücksichtigung ergonomischer Konzeptionen in Anwendungssoftware zu vermindern. Das Programmsystem MENÜGENERATOR gestattet dem Software-Entwickler, interaktiv die Menüstruktur als Menübaum sowie die visuelle Menüabbildung für sein Anwendungsprogramm identisch mit der Sichtweise des Endnutzers festzulegen. Aus der Definition von Struktur und visuellem Interface leitet MENÜGENERATOR dann ein Zielprogramm ab, das als Programmrahmen das definierte Anwendermenü realisiert. In diesen Prozedurrahmen sind dann vom Programmierer die Verarbeitungsprozeduren des Anwenderprogrammes an definierten Stellen einzubinden.

1 Problemstellung

Das Software-Werkzeug MENÜGENERATOR ist vor dem Hintergrund der in jüngster Zeit zunehmenden Diskussion um 'Software-Ergonomie' einzuordnen. Gemeint ist hier eine Berücksichtigung von endnutzerzentrierten Komponenten, beim Software-Engineering generell sowie bei flexiblen Software-Werkzeugen im Einzelnen. In diesem Umfeld sind eine Vielzahl von grundsätzlichen Problemen zu erkennen, die einer prinzipiellen Aufarbeitung bedürfen. Darüberhinaus besteht eine fast unüberschaubare und vor allem schwer strukturierbare Vielfalt von Einzelfragen bei der praktischen Realisation 'ergonomischer' Software mit zeitgemäßen Konzeptionen des Software-Engineering //6/.

Ziel dieses Beitrags ist weniger, grundsätzliche Überlegungen in den Vordergrund zu stellen oder Detailfragen allgemein zu reflektieren. Vielmehr soll ein realisiertes Konzept vorgestellt werden. Darin wurde ein konkreter Versuch unternommen, unter Einbeziehung verschiedener Ideen gleichzeitig zur Verbesserung der Software-Entwicklungsumgebung wie auch der Zielumgebung für den Einsatz der entwickelten Software beizutragen.

Host für die bisher realisierte Version von MENÜGENERATOR sind CP/M-80 Mikrocomputersysteme, sowohl für die Software-Entwicklung wie auch für die generierten Zielprogramme. Konzeption und Software sind jedoch prinzipiell von dieser konkreten Implementierung unabhängig und so allgemein, daß ein Transport in andere Systemumgebungen möglich ist. MENÜGENERATOR selbst ist in Pascal geschrieben und generiert in der vorliegenden Version eine Pascal-Quelle als Zielprogramm. Wie für die Implementie-

rungsumgebung gilt analog für den generierten Zielcode, daß grundsätzlich auch die Generierung von Rahmenprogrammen in anderen höheren Programmiersprachen möglich ist.

Im Hinblick auf das hauptsächlichste Einsatzprofil ist vor allem an die Einsatzfelder von Mikrocomputern als Allzweck-Arbeitsplatzsysteme gedacht, deren normale Betriebsform im Benutzerdialog besteht. Diese Art von Systemen findet zunehmend auch im kommerziellen Bereich Einsatz. Dabei ist regelmäßig davon auszugehen, daß Bedienung wie auch Anwenderdialog von einer Klasse von Benutzern realisiert wird, bei denen hard- und softwarenahe Kenntnisse eher nicht erwartet werden können. Von daher kommt dem Komfort in Bedienung und Handling aus Anwendersicht, der für die Akzeptanz eines Systems mit entscheidend ist, eine besondere Bedeutung zu.

In bezug auf 'Hardware-Ergonomie' hat es in letzter Zeit viele Verbesserungen gegeben, etwa: freibewegliche Tastaturen, reflexionsarme Bildschirme, Geräuschpegelverminderungen, äußeres Gerätedesign, u.v.a.m.. Insbesondere aber hat sich das Bewußtsein über diesen Problemkreis auch in einer Fülle von betriebsrechtlichen Vorschriften niedergeschlagen. Demgegenüber scheint sich Bewußtsein im Hinblick auf Gestaltungsmerkmale der von der Software determinierten Komponenten der Benutzerschnittstelle erst langsam zu bilden. Dies gilt vor allem auch für den Benutzerdialog in kommerziell eingesetzter Software und damit zusammenhängenden psycho-physischen Einwirkungsmustern auf den Menschen /2/. Zu operationalen objektiven Merkmalen in der Beschreibung von Benutzeroberflächen, die Basis einer Normendiskussion und Herleitung eines Normenkonsens' sein könnten, ist es sicherlich noch ein weiter Weg.

Selbst die eher wenigen Systeme, die als Beispiele für gutes Design einer Dialogschnittstelle /5/ angeführt werden können, bieten sehr unterschiedliche Formen der Benutzerschnittstelle. Dies ist einerseits verständlich, da derartige Gestaltungsfragen das schwierige und komplexe Feld menschlicher Verhaltensmuster berühren. Hier ist ohnehin Konsens alles andere als üblich. Andererseits ist die Vielfalt im Design realisierter Lösungen aber auch nicht unbedingt sofort einleuchtend. Unter mehr technischer Sicht kann man nämlich festhalten, daß bei der interaktiven Arbeit mit einem Dialogsystem an der Benutzeroberfläche im wesentlichen immer wieder die gleichen Funktionen abzuwickeln sind /3/. Hierbei geht es insbesondere um:

- Menüwahl und Blättern,
- Statusanzeigen,
- Help-Anforderungen,
- Fehlerkorrekturen,
- Maskenwahl und Vor- sowie Rücksprünge,
- Maskeneingaben, Editieren und Vorbesetzen, sowie darauf aufbauend
- konsistentes Einbetten anwendungsspezifischer und individueller Benutzeroberflächen in die Dialog-Mimik.

MENÜGENERATOR bietet insbesondere für dieses Funktionsbündel im Rahmen der Anwendungsumgebung typischer kommerzieller Anwendungspakete Softwarewerkzeuge. Diese erlauben, die Systementwicklung zu vereinfachen und bieten durch die implementierten Konzepte verschiedene handwerkliche Möglichkeiten, bei der generierten Software ergonomische Aspekte einfließen zu lassen.

Im Hinblick auf Innovationseinschätzung sehen wir die Besonderheiten von MENÜGENERATOR weniger im richtungsweisenden und fundamental neuen Detail. Vielmehr haben wir mehr pragmatisch orientiert einen Versuch unternommen, zu einer möglichst ausgewogenen Abstimmung ganz unterschiedlicher Merkmale und zeitgemäßen Vorstellungen im Softwaredesign zu gelangen, wie etwa:

- anwenderfreundliche Konzepte für Dialoggestaltung im generierten Zielprogramm unter Einbeziehung eines breiten Bündels von Merkmalen in Bereichen wie: flexible Benutzerführung bei Blättern oder direkter Kommando-Anwahl ('Short-Cuts' 73/), (Farb-)Visualisierungsoptionen, Maskenedierung, Fehlerbehandlung, u.a.m.;
- systementwicklerfreundliche interaktive Entwicklungsumgebung;
- lauffähiges und dokumentiertes in Mikrocomputerumgebung realisiertes Softwarewerkzeug zur Generierung von Pascal-Anwendungsprogrammen;
- Berücksichtigung vielfältiger technischer Merkmale zeitgemäßer Softwarearchitektur wie z.B.: CRT-Typ-unabhängig durch Terminaltreiber-Konzept, transportabel in verschiedene Benutzersprachen durch extern edierbare Menütex-te, anpassungsfähig an Strukturänderungen u.a. durch Abbildung des Menübaumes auf ein extern mit Editoren manipulierbares Datenmodell, u.a.m.;
- einsatzbereit für die Entwicklung kommerzieller Anwendungsprogramme bis zu mittlerer Größenordnung mit Passwortintegration;
- in der Konzeption hard- und softwareunabhängig.

Die Strukturierung der weiteren Ausführungen folgt der Sichtweise des Systementwicklers, der MENÜGENERATOR für die Generierung von Anwendungs-Zielprogrammen verwendet. Die dazu komplementären Schichten der Benutzer-/Anwendersicht bei Verwendung der Zielprogramme fließen durch fallweise Erläuterungen ein, wobei insbesondere Akzente aus dem Umfeld ergonomischer Überlegungen betont werden.

2 Ziele und Leistungsmerkmale von MENÜGENERATOR

Die Hauptziele des Programmentwicklungssystems MENÜGENERATOR /1/ sind auf zwei Ebenen zu sehen. Einmal soll dem Software-Entwickler ein leistungsfähiges interaktives Werkzeug für die Abbildung eines Anwendungsprogramms in Menütechnik an die Hand gegeben werden. Dieses Werkzeug unterstützt insbesondere das Design aus Endnutzersicht, Top-down-Architekturen beim Software-Engineering und die Makro-Verzahnung (statische und dynamische Prozedurverknüpfung und -zusammenspiel) komplexer Anwendungsprogramme, denen eine Menüstruktur zugrunde liegt. Die bei komplexen und differenziert verschachtelten Menü-Anwendungsprogrammen für die programmtechnische Realisierung der Benutzer-Menüsicht notwendige oft mühevoll implementierung der (dynamischen) Ablaufstruktur wird durch ein klares Prozedur-Aufrufkonzept abgebildet.

Zum zweiten werden für die Anwendungsprogramme selber durch die Verwendung von MENÜGENERATOR Standardhilfsmittel für Menügestaltung, Menüablauf und Maskeneingaben angeboten, die vielfältige Vorstellungen ergonomischer Softwaregestaltung an der Nutzeroberfläche berücksichtigen.

So sind zum Beispiel in MENÜGENERATOR selber wie auch in den erzeugten Zielprogrammen Merkmale visualisierter Benutzerführung, gegebenenfalls unter Einschluß von Farbe, wie: Ge-

staltungsoptionen des Menübaums (Auswahlmenü-Position, Überlagerung, Löschung von hierarchisch übergeordneten Menüs, Scroll-

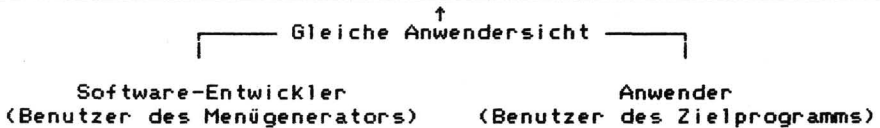
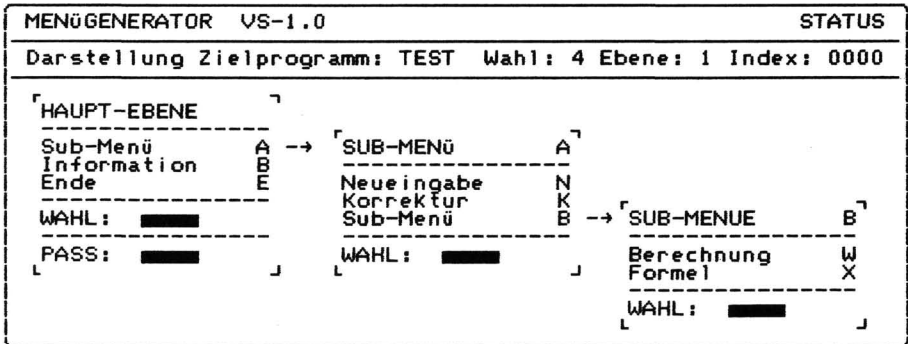


Abb.1: Menüdarstellung

ling der Auswahlalternativen, Passwortschutz vor Menüanwahl), Wahl visueller Attribute (Invers-Video, Farbe), Wahlstrategien für ungeübte und geübte Benutzer (schrittweises Durchblättern, Direktanwahl), Help-Informationen, lückenlose Statusinformationen und endnutzer-ausgerichtete intuitive Masken-Editfunktionen realisiert. Der Anwendungsprogrammierer kann in der Entwurfsphase den Menübaum in seiner Struktur festlegen, die notwendigen Menütexte hinzufügen und insbesondere beliebige Erweiterungen oder Korrekturen im Hinblick auf Struktur und Texte durchführen. Das Bildschirmlayout für die Dialogführung des generierten Anwendungs-Programmrahmens und die visuellen Attribute im Detaildesign sind unmittelbar am Bildschirm nachvollziehbar (Abb.1).

Der Menügenerator erzeugt als Output ein Pascal-Zielprogramm. Die höhere Programmiersprache Pascal wurde u.a. deshalb gewählt, weil sie vielfältige Strukturierungskonzepte unterstützt, vergleichsweise hohe Portabilität besitzt und in Mikrocomputerumgebung mit leistungsfähigen Compilern und Software-Entwicklungswerkzeugen verfügbar ist.

Das generierte Zielprogramm enthält alle Prozeduren für Menüauswahl und Steuerung, sowie Kommentartexte an Stellen im Zielprogramm, an denen noch Programmcode eingefügt werden muß. Das Zielprogramm ist fehlerfrei übersetzbar, aber es bildet allein das Skelett eines Anwenderprogrammes. Die Funktionen des Anwenderprogrammes müssen mit Hilfe eines Editors in der Form von Prozeduren nachträglich eingefügt werden.

3 Benutzerschnittstelle des MENÜGENERATORS

Das Design der Benutzerschnittstelle stellt den Kern des entwickelten Programmkonzeptes dar. Wie haben versucht, MENÜGENERATOR anwenderzentriert zu entwickeln und eine Übertragung dieser Sichtweise auch für die zu generierenden Anwendungsprogramme zu erzwingen. Dazu gehört an vorderer Stelle, daß die Menue-Darstellung für den Benutzer des MENÜGENERATORS und den Zielprogramm-Anwender in der Sichtweise identisch ist (siehe Abb.1).

Beim vorliegenden interaktiven System zur Menügenerierung wurden als Rahmenbedingungen verschiedene Anforderungen realisiert, die Gegenstand softwareergonomischer Überlegungen sind /8/:

Bildschirmaufteilung	- fixe Bereiche auf dem Bildschirm für Fehler-, Info-, Statusmeldungen.
Steuerungs- und Transaktionsmöglichkeiten	- Programmsteuerung mit Abbruchmenü, Statusanzeigen, Eingabe-Infos anfordern, Benutzereingaben über Maskenfelder.
Visualisierung der Informationsdarstellung	- Invers Video, helle / halbhelle Zeichendarstellung, blinkende Zeichendarstellung, 8 Vorder-/Hintergrund-Farben.

3.1 Dialog-Gestaltung

Das System gibt Hinweise, Texte und Eingabeanforderungen auf einheitlichen Bildschirmzonen aus, und erwartet in der vorliegenden Version Eingaben des Benutzers über die Tastatur. Konzeptionell ist die Integrierung eines Lichtgriffels für eine Teilmenge der Menüwahlkonzepte möglich. Die einzugebenen Daten und Anweisungen werden immer in formatierte Felder eingetragen. Dabei wird jedes Zeichen unmittelbar auf Zulässigkeit geprüft und dann angezeigt. Zusätzlich werden Hinweise über Art und Umfang der geforderten Eingabedaten auf dem Bildschirm angezeigt. Solche Meldungen (kurz: was macht das System im Augenblick und was darf der Benutzer eingeben?) werden von MENÜGENERATOR bei allen Eingaben grundsätzlich angezeigt. Das System ist so konzipiert, daß fehlerhafte Eingaben so weit wie irgendmöglich abgefangen werden und nicht zum Programmabbruch führen. Bei unzulässigen Eingaben erhält der Benutzer eine Fehlermeldung.

Im weiteren folgt eine Erläuterung der wesentlichen Merkmale der Dialog-Gestaltung.

Statusanzeige. Um den Benutzer bei Verarbeitungsfunktionen nicht im Unklaren über den Systemzustand zu lassen, gibt das System grundsätzlich eine aktuelle Anzeige über den Systemstatus in benutzerorientierter Sicht an einer fixen Bildschirmposition aus.

Mit benutzerorientierter Sicht ist der Versuch gemeint, Systemabarbeitungsfunktionen grundsätzlich auf die Oberfläche des für den Benutzer transparenten Anwendungsgebietes abzubilden (etwa: "Sortierung Artikelstammdatei", und nicht "Aufruf Prozedur SORT von UPDATE-Programm"). In der rechten oberen Bildschirmcke erscheinen derartige Meldungen bei zeitintensiven Verarbeitungen, Eingabeanforderungen, Eingabefehlern oder der Anzeige des Abbruchmenüs. Kurzmeldungen wie 'WARTEN', 'FEHLER', 'ABBRUCH', 'EINGEBEN', 'WÄHLEN' erfolgen unter Verwendung von farbigen, blinkenden oder invers-video Attributen.

Es wurde darauf geachtet, Benutzerreaktionen im normalen Verlauf des Anwendungsdialogs nicht durch blinkende Meldungen abzuverlangen. An der Bildschirmoberfläche liegt in der Verwendung von Blinkattributen eine hohe potentielle Stressquelle. Sicherlich ist dies eine wichtige Komponente im Rahmen des Themenfeldes Softwareergonomie.

Infotext. Unmittelbar nach Start des Menügenerators wird dem Benutzer zunächst eine Erläuterung zum Programm angeboten. Auf mehreren Bildschirmseiten werden Informationen, Begriffsdefinitionen und Funktionen beschrieben. Zusätzlich kann bei jeder Maskenfeldeingabe ein sogenannter 'Infotext' angefordert werden. Bei vielen Maskenfeldeingaben reicht die Information des Maskenvortextes möglicherweise nicht aus. Der Benutzer kann dann an beliebiger Eingabeposition die Taste <?> betätigen. Es erscheint der entsprechende Infotext zu diesem aktuellen Eingabefeld. Nach zwei weiteren Tastenfunktionen wird der spezielle Hinweis wieder gegen den Standard-Infotext ausgetauscht. Der Standardtext beschreibt die notwendigen Steuertasten, um Gruppen von Eingabefeldern zu edieren. Allerdings sind auch die Maskenvortexte mit Kurzhinweisen auf die erlaubten Eingabetasten versehen, so daß der geübte Anwender meist auf die Hilfs-Informationen verzichten kann.

Fehlerbehandlung. Fehlerhafte Eingaben erfordern bei MENÜGENERATOR zumeist Benutzerreaktionen ("interaktive und kommunikative Auflösung des Mißverständnisses") und werden regelmäßig nicht automatisch geheilt. Dies gilt vor allem für typische Kommunikationsmißverständnisse im Mensch-Maschine Dialog, z.B. Verwechslung von <Ziffer-Null> und <Buchstabe O>, Buchstabeneingabe in numerische Felder, Funktionsabrufversuch mit undefinierten Kommandos, u.a.m.. Wir halten diese kommunikative Fehlerbehandlungsstrategie für unabdingbar in Anwendungsprogrammen. Semantische Fehleranalyse ist natürlich kontextabhängig vom Anwendungsproblem. Ein Software-Werkzeug, wie MENÜGENERATOR, kann nur standardisierte Mechanismen zur Unterstützung bereitstellen und formale Zeichen- und/oder Zeichenkettenabprüfungen unterstützen. Letzteres ist bei MENÜGENERATOR lückenlos für alle Ebenen der Menüansteuerung und Masken-Eingabe realisiert.

Konkret erfolgen typischerweise bei einer nicht definierten/erlaubten Eingabe folgende Reaktionen:

- akustisches Signal
- Statusanzeige FEHLER
- an der Position der Fehlererkennung ein '?'
- Fehlererläuterung

Der Fehlertext gibt Auskunft über die Art des erkannten Fehlers (z.B. eine nicht zulässige Eingabetaste). Nach dem Betätigen der <ESC>-Taste stellt das Programm den ursprünglichen Zustand wieder her:

- der Fehlertext und das '?' werden gelöscht
- der Cursor wird wieder an die Stelle bewegt, an der der Fehler erkannt wurde.

Benutzereingaben. Der Menügenerator verwendet im wesentlichen zwei Eingabetechniken, und zwar Eingabemasken und das

Abbruchmenü zur Programmverzweigung.

Bei den Eingabemasken werden die Daten in formatierte Felder eingetragen. Dabei wird jedes Zeichen unmittelbar auf Zulässigkeit im Sinne der Zugehörigkeit zum jeweils definierten Zeichenalphabet der Maske geprüft und dann angezeigt. Die maximale Stellenzahl ergibt sich aus der Länge des Feldes, der Zulässigkeitsbereich aus dem Maskenvortext sowie durch den zugehörigen Infotext.

Die Menütechnik ist vorteilhafter bei einer größeren Anzahl von zulässigen Auswahlmöglichkeiten. Sie wurde daher zur Programmsteuerung beim Abbruchmenü in MENÜGENERATOR eingesetzt, das als Interrupt an beliebiger Stelle im Dialog aufrufbar ist und eine Verzweigung in alle Systemfunktionen erlaubt.

CURSOR → TASTE ↓	Cursor an 1. Stelle des Feldes	Cursor an 2. - vorletzter Stelle des F.	Cursor an letzter Stelle des F.
<CTRL-A>	Abbruch alter Inhalt bleibt	Abbruch, bis hier eingegeb. Werte überschreiben alten Inhalt	
--> oder <CTRL-D>	Cursor um eine Position nach rechts, alter Inhalt bleibt		zum nächst. Feld, Inhalt bleibt
<-- oder <CTRL-S>	zum Feld vorher 1. Position	Cursor um eine Position nach links	
<RETURN> oder <CTRL-X>	zum nächst. Feld, alter Inh. bleibt	bis hier eingegebene Werte werden übernommen, dann weiter zum nächsten Feld	
<zulässig. Eingabe- Zeichen>	Zeichen überschreibt, Cursor um eine Position nach rechts		überschreibt, Cursor bleibt stehen

Abb.2: Masken-Edieren mit Steuertasten

Bei beiden Eingabetechniken erhält der Benutzer unmittelbar nach dem Drücken einer Taste ein 'Feedback' auf dem Bildschirm. Dies kann die Bestätigung und Annahme des empfangenen Zeichens durch Echo-Darstellung sein oder eine Fehlermeldung mit Korrekturhinweisen. Bei der Eingabe von Steuerzeichen erfolgt eine Bewegung des Cursors oder die Ausgabe einer Systemmeldung.

Neben der Eingabe von zulässigen Zeichen kann der Benutzer auch Steuertasten betätigen oder Control-Zeichen eingeben. Dies ermöglicht ein Edieren innerhalb des Eingabefeldes, um Korrekturen oder Änderungen durchzuführen. Die Steuertasten sind auf der Terminaltastatur auf die üblichen Cursor-Tasten, z.B. → und ←, abgebildet. Die Ausführung einer Cursorbewegung innerhalb eines Eingabefeldes ist von der aktuellen Cursorposition abhängig. Normalerweise bewirken die Tasten → oder <CTRL-D> eine Bewegung des Cursors um eine Position nach rechts und die Tasten ← oder <CTRL-S> die Positionierung nach links (zur implementierten Logik siehe Abb.2).

Bei jeder Maskenfeldeingabe kann an beliebiger Feldposition durch das Betätigen der CTRL-A Taste ein genereller Interrupt durch Anforderung des Abbruchmenüs durchgeführt werden, um die Hauptfunktionen des Systems zu wechseln.

Die Control-Tasten sind wegen der Kompatibilität zu bestehenden und weitverbreiteten Standard-Softwarepaketen der Mikrocomputerumgebung (z.B. WORDSTAR, DATASTAR usw.) alternativ einzusetzen. Bei diesen Paketen erfolgt, zusätzlich zu Funktions-tastenbelegungen, oft eine Abbildung notwendiger Benutzeraktivitäten auf Tastenmuster normaler alphanumerischer Tastaturen (z.B.: <CTRL-E>, <CTRL-S>, CTRL-D) und <CTRL-X> bedeuten: nach oben, links, rechts und unten; kontextgebunden <CTRL-A> und <CTRL-F> entsprechend: weit-nach-links und weit-nach-rechts) /9/. Wir sehen derartige Konzeptionen für geübte Anwender als sehr intuitiv und mächtig an.

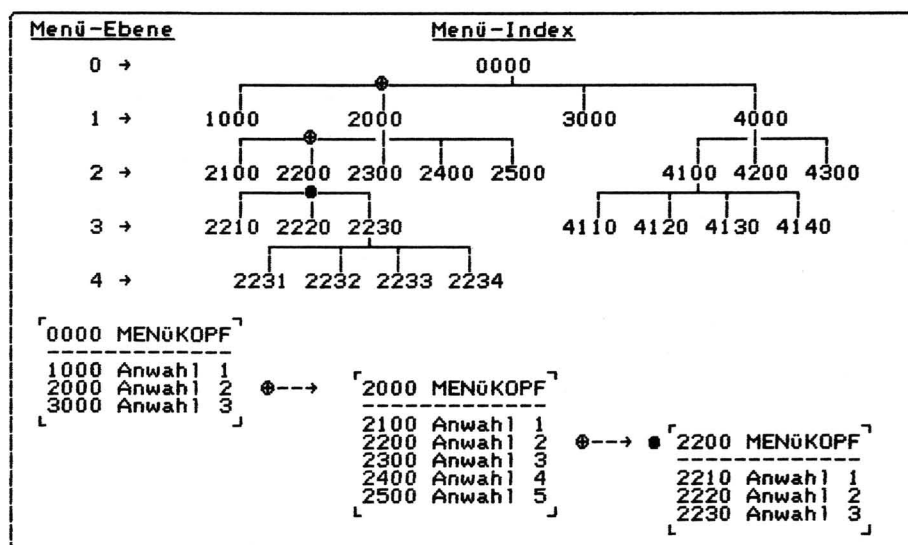


Abb.3: Menüstruktur

Der Benutzer kann innerhalb einer logisch zusammengehörigen Maskengruppe beliebig die Eingabefelder wechseln und Korrekturen vornehmen. Bei jedem Neuedieren eines sichtbaren Maskenfeldes erscheint zuerst als Default der alte Maskeninhalte.

3.2 Systemfunktionen

MENÜGENERATOR stellt folgende Haupt-Funktionen bereit:

- Interaktive Eingabe der Menüstruktur (mit Korrekturmöglichkeiten).
- Eingabe von Menütexen mit Darstellungsattributen, Parametern und Passworten.
- Diskettenspeicherung von Menüstruktur und Menütexen.
- Darstellung und Anwahlsteuerung der Menüs.
- Generierung des Zielprogramms mit Prozeduren zur Darstellung und Anwahlsteuerung.

Im weiteren folgen Erläuterungen zur Menüdarstellung und Anwahltechnik aus der Sicht des Benutzers.

Menüstruktur. Die hierarchische Anordnung von Menüs bildet typischerweise eine Baumstruktur. Von einem Hauptmenü ausgehend stellen Anwahlzeilen jeweils die Kopfzeilen für neue untergeordnete Menüs in tieferen Ebenen der Struktur dar (Abb.3).

Menü-Darstellung. Die Darstellung eines Menüs kann über bestimmte Parameter gesteuert werden. Es wird nach Darstellungsformen und Darstellungsattributen unterschieden. Die Darstellungsform bezieht sich auf den Aufbau des Menüs mit fixen oder scrollenden Anwahlzeilen. Die Attribute zur Darstellung beeinflussen die Farbgebung, die Textbreite und die Anzeigeposition (Abb.4).

- Fixmodus:** Das Menü wird mit Kopfzeile und allen Anwahlzeilen, sowie der darunterliegenden Eingabezeile auf dem Bildschirm ausgegeben. Es sind maximal 9 Anwahlzeilen möglich, die Textbreite kann wahlweise 10 oder 16 Zeichen betragen.
- Scrollmodus:** Wenn im Anwenderprogramm nur wenig freier Bildschirmraum zur Verfügung steht, kann der Scrollmodus sinnvoll sein. Dies bedeutet, daß die einzelnen Anwahlzeilen jeweils gegeneinander ausgetauscht werden. Dies kann vorwärts (Anwahlzeile 1 -> 9) oder rückwärts (Anwahlzeile 9 -> 1) erfolgen.

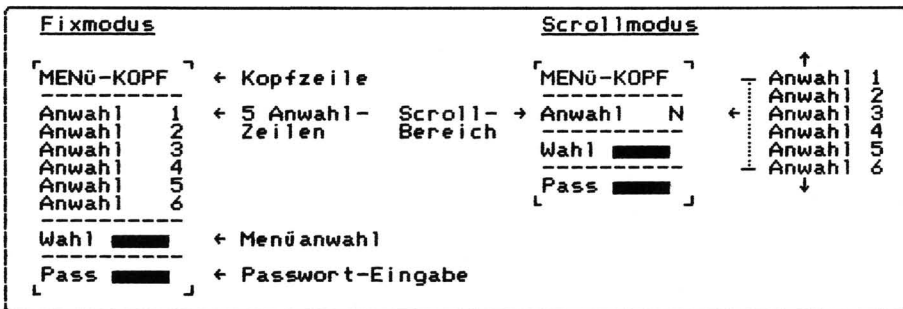


Abb.4: Darstellungsmodus

Darstellungsattribute. Die Attribute zur Darstellung sind die Textbreite, die Farbgebung und die Anzeigeposition am Bildschirm. Jede Kopf- oder Anwahlzeile eines Menüs kann durch inverse, halbhelle oder farbige Darstellung hervorgehoben werden.

Anzeigestatus. Der Anzeigestatus bezieht sich auf die Darstellung eines kompletten Menüs in Abhängigkeit von der aktuellen Menüebene. Normalerweise bleiben die zuletzt in den übergeordneten Ebenen angewählten Menüs auf dem Bildschirm erhalten, sofern sie nicht vom Anwenderprogramm oder nachfolgenden Menüs überlagert werden. Durch einen gesetzten Anzeigestatus (1-4) können hierarchisch übergeordnete Menüs automatisch gelöscht werden.

Menüanwahl. Die Anwahl einer Menüzeile oder eines bestimmten Menüs kann auf verschiedene Weise erfolgen.

Bei einem normalen Menü im Fixmodus erfolgt die Anwahl durch das Eingeben des Anwahlzeichens. Die Eingabe eines ein-

stelligem Anwahlzeichens muß mit der Return-Taste abgeschlossen sein. Bei der Eingabe eines Zeichen, das nicht im Menü aufgeführt ist, erfolgt keine Reaktion. Der Benutzer muß neu eingeben. Es läßt sich sicherlich darüber streiten, ob im weichen Umfeld dessen, was bei Kommunikation als "Fehler" zu definieren ist, konkret an dieser Stelle eine Systemreaktion auf eine Benutzeranfrage über nicht definierte Funktionszeichen-Eingabe ignoriert oder als aufzulösender fehlerhafter Systemzustand anzusehen ist. Wir haben hier z.Zt. ein Systemverhalten im Sinne von 'Sturheit' implementiert, sind aber (auch vor dem Hintergrund der o.a. Fehlerbehandlungsstrategien) nicht sicher, ob dies die 'beste' Lösung ist.

Bei einem Menü, das im Scrollmodus dargestellt ist, können die Anwahlzeilen mit den Tasten → vorwärts und ← rückwärts gescrollt werden. Die Anwahl erfolgt durch das Drücken der Return-Taste. Die aktuell sichtbare Anwahlzeile ist dann angewählt.

Wenn ein Anwenderprogramm viele Funktionen enthält und dementsprechend eine komplexe Menü-Baumstruktur existiert, so müßte sich auch ein geübter Benutzer normalerweise bei jeder Sitzung durch den gesamten Menübaum durcharbeiten. Dieses ist ermüdend und kostet Zeit. Daher ist es bei den durch den Menügenerator erzeugten Menüs möglich, unmittelbar die Funktionsebene anzusteuern, auf der der Anwender arbeiten möchte. Durch die Eingabe eines vierstelligen Menüindex oder Kommandowortes kann jede Anwahlzeile des gesamten Menübaumes direkt erreicht werden. Das Kommandowort stellt eine beliebige vierstellige alphanumerische Abkürzung für eine gewünschte Funktion dar. Hier ist explizite Fehlerauflösung bei Verwendung eines nicht definierten Kommandowortes (natürlich) notwendig.

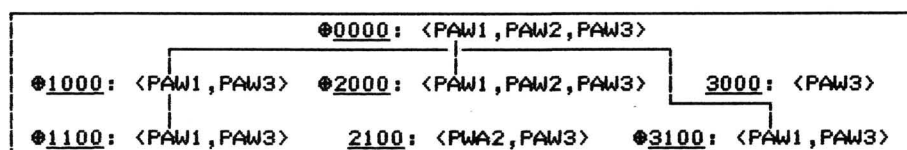


Abb.5: Beispiel für Zugriffsberechtigungs-Baum bei Passwort 'PAW1'

Passwortverarbeitung. Jede Menüanwahl kann durch maximal drei alternative Passwörter, die jedes für sich den Funktionseinstieg eröffnen, geschützt werden. Ein Passwort besteht aus bis zu vier alphanumerischen Zeichen. Wenn nur Leerzeichen im Passwortfeld stehen, so ist kein Passwort angelegt. Falls die Passwörter zwischen hierarchisch unmittelbar aufeinanderfolgenden Menüs übereinstimmen, so entfällt eine Passwortabfrage beim Submenü. Auf diese Weise können ganze Baumzweige über ein Passwort erreicht werden. Durch diesen Mechanismus mit drei Passwörtern lassen sich durch entsprechende Vergabestrategien differenzierte Funktions-Zugriffspfade für die Anwendungsumgebung aufbauen (siehe Beispiel Abb.5). In diesem Beispiel hat etwa der Inhaber von Passwort 'PAW3' Zugang zu allen Menüzweigen, Zugang für 'PAW1' zu Submenü 3100 ist nur über Direktanwahl möglich. Damit sind verborgene Ober-Menüs implementierbar.

4 Basismaschine

MENÜGENERATOR ist auf einem CP/M-Mikrocomputer /10/ implementiert und lauffähig bei Vorhandensein der heute weit verbreiteten Standard-Konfiguration: 64 KB-Hauptspeicher, 2 Diskettenlaufwerke mit mindestens 400 KB-Speicherkapazität und einem Video-Terminal mit unterschiedlichen visuellen Attributen als Konsole. Die Software besteht aus einem komplexen PASCAL/Z-Programm-System, daß sich aus sechs Teilprogrammen zusammensetzt, die jeweils in separate Moduln unterteilt sind. Der erzeugte Zielcode in der realisierten Grundversion VS-1.0 von MENÜGENERATOR wird ebenfalls in Pascal als ASCII-Quellcode generiert und ist direkt lauffähig in der Umgebung des PASCAL/Z-Systems /4/. Durch die Implementierung in CP/M-80 ist es z.B. möglich, eine der auf dem CP/M-Softwaremarkt angebotenen 'Shell'-Konzeptionen zu nutzen, die über den primitiven CP/M-Kommandoprozessor gleichfalls ein menügesteuertes Dialogkonzept (hier: für CP/M-Funktionen) legen (z.B. /7/). In dieser Form lassen sich mit MENÜGENERATOR abgeschlossene turn-key Anwendermaschinen auf durchgängiger Menübasis aufbauen.

MENÜGENERATOR ist portabel in andere Pascal-Compiler-Umgebungen. Weiterhin ist das System so strukturiert, daß eine Generierung des erzeugten Zielprogramms in einer anderen Sprache als Pascal prinzipiell realisiert werden kann. Das zugrunde liegende Datenmodell für MENÜGENERATOR ist derart gestaltet, daß bestimmte Parameter (wie Steuerzeichen-Tabellen für Terminaltreiber), Menüstruktur, Menütex te und die Texte für die Menüführung von MENÜGENERATOR (Eingabefelder, Ausgabete xte) selber auf externen Textdateien gehalten werden. Diese Architektur erlaubt vielfältige schnelle Informationen, Korrekturen und Anpassungen mit Standard-Editoren. Dadurch wird z.B. auch parallele Softwareentwicklung für verschiedene Anwender-Landessprachen erleichtert. Die visuelle Gestaltung von Maskeneingaben (Maskenleerzeichen, Invers-Video, Farbe, Begrenzungszeichen, Default-Inhalt überschreibt) kann durch Parameter gesteuert werden.

5. Schlußbemerkungen

Bei der realisierten Basisversion VS-1.0 des erläuterten Systems handelt es sich um ein Software-Entwicklungs-Werkzeug für den Anwendungsprogrammierer, das eine Vielzahl von Erweiterungsmöglichkeiten ermöglicht:

Mögliche Perspektiven sind:

- Die Generierung spezieller Anwenderfunktionen schon aus MENÜGENERATOR heraus, wie z.B. Maskenfestlegungen;
- Verbesserungen in der dynamischen Menüarbeitungsstruktur, wie z.B. Fixierung und direkte Rücksprungmöglichkeit auf schon erreichte Menüweige;
- Generierung von Hilfsfunktionen im Zielprogramm.

Die Generierung von Help-Funktionen für jegliche Menüanwahl wird zwar prinzipiell durch MENÜGENERATOR durch explizite Aufnahme in

die Menüs der Zielprogramme unterstützt. Die bisherigen Erfahrungen haben jedoch gezeigt, daß ein impliziter Mechanismus, analog zu dem in MENÜGENERATOR selber implementierten, auch für die generierten Zielprogramme bereitgestellt werden sollte. Diese und andere Erweiterungen sind für die in Arbeit befindliche Version VS-2 von MENÜGENERATOR vorgesehen.

Man könnte weiter daran denken, die durch MENÜGENERATOR auf der Makroebene unterstützte Programmentwicklungsumgebung durch integrierte weitere Werkzeuge für viele Details der Softwareentwicklung zu verbessern (z.B. Maskeneingabe-Generator, Reportgenerator).

Literaturhinweise

- /1/ Bernartz, W.:
Realisierung eines Menügenerators. Benutzerhandbuch.
Arbeitspapier Schwerpunkt Wirtschaftsinformatik & OR. Hrsgb.
L. Nastansky. Fachbereich Wirtschaftswissenschaften, Univer-
sität Paderborn. Paderborn 1982, S. 1-56.
- /2/ Dean, M.:
How a computer should talk to people.
In: IBM Systems Journal, Vol. 21 (1982) 4, S. 424-453.
- /3/ Martin, J.:
Design of Man-Computer Dialogues.
Series in Automatic Computation. Englewood Cliffs NJ (Pren-
tice-Hall) 1973, insbes. S. 111-124.
- /4/ Moskow, J.:
PASCAL/Z Manual (Version 4.0).
Ithaca Intersystems, Ithaca NY 1982.
- /5/ Smith, D.C., C. Irby, R. Kimball, B. Verplank:
Designing the Star User Interface.
In: BYTE, Vol. 7 (1982) 4, S. 242-282.
- /6/ Schnupp, P., C. Floyd:
Software, Programmentwicklung und Projektorganisation.
2. Auflage, Berlin-New York (de Gruyter) 1979.
- /7/ Supervyz:
User's guide and reference manual.
Epic Computer Corp., o.O. 1981.
- /8/ Weiss, H.:
Windigen Programmierern das Handwerk legen.
In: Computer Magazin, 11.Jg. (1982) April, S. 86-90.
- /9/ WordStar:
User's guide.
Version 3.0. San Rafael CA (Micropro) 1981.
- /10/ Zaks, R.:
CP/M handbook with MP/M.
o.O. (Sybex) 1980 (deutsche Ausg. 1981).

o.Prof. Dr. L. NASTANSKY / Dipl.-Inf. W. BERNARTZ
Schwerpunkt WIRTSCHAFTSINFORMATIK & OR
UNIVERSITÄT GH PADERBORN
Warburger Str. 100, D-4790 Paderborn