KONZEPTE FÜR UND ERFAHRUNGEN MIT DPF, EINEM MASKENORIENTIERTEN EINHEITLICHEN BILDSCHIRM- UND DRUCKER- EIN-AUSGABE SYSTEM

H. H. Adelsberger, Wien

Zusammenfassung: Ein maskenorientiertes einheitliches Bildschirm- und Druckereinausgabesystem wird präsentiert. Schwerpunkt des Designs sind sowohl die ergonomischen Aspekte der Benutzung wie auch diejenigen der Softwareerstellung. Im speziellen wurde bei DPF auf Einheitlichkeit der zugrundeliegenden Konzepte, auf Portabilität und Flexibilität geachtet.

1 Einleitung

In der vorliegenden Arbeit präsentieren wir DPF (Dataentry and Print Form Package), ein Softwarepaket zur Handhabung der Bildschirm- und Druckerschnittstelle von Softwaresystemen und Programmen. Subsysteme dieser Art gehören in den Bereich der Mensch-Maschine-Schnittstelle.

Obwohl die ersten Publikationen zu Problemen dieser Art bereits Ende der fünfziger Jahre zu finden sind und derzeit eine Vielzahl von Publikationen zu diesem Thema teils in der EDV-Fachliteratur, teils in den Publikationsmedien der betroffenen Disziplinen (z.B. Psychologie) erscheinen, hat dieser Bereich in der (akademischen) Forschung nicht so viel Aufmerksamkeit gefunden wie etwa Arbeiten im Bereich von Algorithmen, Computersprachen und Datenbanksystemen. Für den kommerziellen Bereich gilt dies nur in wesentlich eingeschränkterem Maß: Am Anfang der Datenverarbeitung stand die effiziente Ausnützung des Prozessors und der Datenspeicher im Vordergrund. Aber bereits die ersten höheren Programmiersprachen (FORTRAN und COBOL) reflektierten die Notwendigkeit der Kommunikation zwischen Rechner und Außenwelt: "formatierte" Ein-Ausgabe in FORTRAN, Picture Klausel und Report Writer in COBOL. Charakteristisch jedoch: ALGOL60, die

"wissenschaftliche" Computersprache, verzichtete auf die Definition der Ein-/Ausgabe. Mit dem Aufkommen der ersten interaktiven Systeme änderte sich die Situation schnell. Die Notwendigkeit, wesentlich mehr Aufmerksamkeit beim Systemdesign auf die Benutzer des Systems zu legen, wurde schnell erkannt. Insbesondere die großen Computerhersteller boten für ihre Hardware und Operatingsysteme entsprechende Unterstützung an.

Es bestehen nun mehrere Gesichtspunkte, unter denen die Mensch-Maschine-Schnittstelle betrachtet und behandelt werden kann. Wir wollen hier die ergonomische Seite der Erstellung und Wartung von Systemen, bei denen dieser Bereich von größerer Bedeutung ist, in den Vordergrund stellen (andere Aspekte wären z. B. Hardware oder "human factors"). Wenn die Belastung des Programmierers und System Designers gering gehalten und er von unnötigem Ballast frei gehalten wird, er einfache, aber mächtige Werkzeuge zur Hand hat, steigt die Produktivität, aber auch die Qualität der geschaffenen Softwareprodukte.

Untersucht man typische interaktive Systeme, wie z. B. Management-, Personal- und Büroinformationssysteme, so erkennt man klar, daß ein Großteil des Arbeitsaufwands bei der Softwareerstellung und -wartung im Bereich der Mensch-Maschine-Schnittstelle zu finden ist. Leistungsfähige Unterstützung in diesem Bereich zur Verfügung zu haben, hat daher zentrale Bedeutung für die Entwicklung solcher Systeme.

Wir sind weiter davon überzeugt, daß in der Zukunft der Standard-Anwender-Software noch wesentlich größere Bedeutung zukommen wird als derzeit. Daß sich Standard-Anwender-Software momentan noch nicht so stark durchgesetzt hat, liegt wohl daran, daß diese Systeme zu wenig flexibel sind, um den Ansprüchen der verschiedenen Benutzer, wie auch den sich immer wieder ändernden Gegebenheiten voll gerecht zu werden. Für Systeme, die allen diesen Forderungen entsprechen, führten wir die Bezeichnung "Individuelle Standard-Anwender-Software" ein. Einer der Angelpunkte für solche Systeme ist eine möglichst flexible Mensch-Maschine-Schnittstelle.

Damit wäre die Problemstellung umrissen. Vor etwa zwei Jahren begannen wir die vorhandenen Systeme zu untersuchen, um aufbauend auf den Erkenntnissen ein neues, leistungsfähiges Support Package zu erstellen. Als Vorteil erwies sich hierbei, daß wir sowohl Erfahrungen mit von uns selbst entwickelten Supportsystemen (einem Masken unterstützenden System und einem Report Writer /1/) hatten, wie auch mit einer Vielzahl von Anwenderprogrammen aus wissenschaftlichen und kommerziellen Bereichen.

2 Das Design

Die Erkenntnisse und Schlußfolgerungen dieser Analyse führten zur Formulierung von vier Hauptpunkten, die die Basis des Designs ausmachen:

2.1 Einheitlichkeit

Es besteht in vielerlei Bereichen eine Ähnlichkeit zwischen der Behandlung der Masken-Ein-Ausgabe und der gedruckten Ausgabe. Wir legten fest, daß diese beiden Bereiche mit demselben Supportsystem bearbeitet werden sollen und nur dort, wo inhaltliche Unterschiede bestehen, unterschiedliche Behandlung erfolgen soll. Dies ermöglicht etwa, Bildschirmmasken am Drucker oder Formulare am Schirm auszugeben.

Hat die Mensch-Maschine-Schnittstelle aus der Warte des Benutzers ihre ganz spezifischen Aspekte ("humen factors"), so stellt sie aus der Warte der Software-Erstellung nur eine der beiden großen Bereiche der Ein-Ausgabe dar. Der andere ist der Austausch von Daten zwischen Programmen und Files bzw. Datenbanken. Von dieser Warte aus besteht eine große Anzahl von Ähnlichkeiten und strukturellen Verwandschaften. Ein leistungsfähiges Supportsystem sollte diese Ähnlichkeiten berücksichtigen. In DPF ist dies derzeit noch nicht der Fall. Die Schnittstelle eines relationalen Datenbanksystems, das derzeit von uns entwickelt wird, wird jedoch weitgehend auf DPF abgestimmt, sodaß sowohl der Datenaustausch zwischen Programm und Mensch als auch

mit dem Datenbanksystem gleichartig abläuft.

2.2 Portabilität

Mit Ausnahme von wenigen Systemen waren die maskenorientierten Systeme extrem Hardware- und Operatingsystem-abhängig. Die mangelnde Portabilität mag zwar im Interesse der Computerhersteller sein, nicht jedoch im Interesse der Anwender. Daß auf diesen Punkt noch immer so wenig geachtet wird, liegt wohl daran, daß die negativen Auswirkungen von falschen Entscheidungen erst mit großer Verzögerung wirksam werden. Wir legten fest, daß diesem Bereich große Aufmerksamkeit geschenkt wird:

- verschiedene Hardwarekonfiguration / gleiches Betriebssystem: Alle hardwarespezifischen Informationen sind auf einer Datei gespeichert, von der die Anwenderprogramme die notwendigen Informationen bei Programmbeginn einlesen. Wird die Hardware getauscht (Schirm oder Drucker), so muß nur diese Information adaptiert werden, was in einfachster Weise erfolgt; das Anwenderprogramm braucht nicht neu übersetzt werden.

- verschiedene Betriebssysteme: DPF ist fast vollständig in einer höheren Programmiersprache geschrieben; implementationsspezifischer Code ist auf ein absolutes Mindestmaß beschränkt. Derzeit ist DPF in FORTRAN vorhanden, an einer PL/I-, COBOL- und PASCAL-Version wird derzeit gearbeitet; eine ADA-Version ist in Vorbereitung. DPF existiert in zwei verschiedenen Fassungen: Für Betriebssysteme mit Blockmodetransfer und für Betriebssysteme mit Einzelzeichenkontrolle. Diese Unterscheidung hat jedoch keine Auswirkung für den Programmierer; die Unterprogrammspezifikationen sind in beiden Fällen dieselben.

- Maskenbibliothek: Eine der wesentlichsten Richtlinen war, daß im Gegensatz zur meist interaktiven Erzeugung einer Maske in anderen Systemen in DPF die Definition der Maske über ein gewöhnliches Editor File erfolgt. Dies hat den großen Vorteil, daß eine Kopie der Maskenbibliothek in Form eines simplen sequentiellen Files auf andere Rechner gebracht werden kann.

2.3 Flexibilität

Die Flexibilität kann als größter Vorteil von DPF angesehen werden. Diese Flexibilität wird durch zwei Prinzipien erreicht:

(a) Alle die Maske bestimmenden Charakteristika sind komplett unabhängig vom Programm, das die Maske benützt. Dies betrifft sowohl die Hintergrundinformation (d.i. jener Teil der Maske, der unveränderlich ist), die Felder der Maske, in denen die Variablen angezeigt oder in die sie eingegeben werden, wie auch die Attribute der Felder (read-only / read-write / write-only; blink / no blink; usw.). Dies bedeutet, daß eine Maske in jedwedem Sinn abgeändert werden kann, ohne daß das benutzende Programm neu übersetzt werden muß.

(b) Die Maske entscheidet (und nicht das Programm!), welche Variablen zwischen dem Menschen und der Maschine ausgetauscht werden. Dies hat zur Konsequenz, daß der Umfang des Datentransfers von der Maske aus bestimmt werden kann. Maskenfelder können zusätzlich auf eine Maske gebracht werden oder von ihr gelöscht werden, wieder, ohne daß das Programm geändert werden muß.

Erreicht wird dieses Design durch ein Prinzip, das einen ausgesprochen komfortablen Programmierstil ermöglicht: Variablen des Programms und Felder einer Maske werden mittels Nummern, sogenannten "logischen Nummern", referenziert. Die Verbindung zwischen den Variablennummern und den Variablen wird einmal für ein Software-System hergestellt (in der Programminitialisierungsphase). Dann genügen zwei einfache Prozeduraufrufe SHOWF (show_form) und READT (read_terminal), um u.a. folgende Aktionen zu veranlassen: Anzeigen der Maske am Schirm; Anzeigen der Maskennummer in der Statuszeile; Positionierung des Cursors auf das ersten Feld; Unterstützung der Cursorbewegung (links / rechts / hinauf / hinunter, Tabulatorsprung: vorwärts / rückwärts / erstes Feld/ letztes Feld); Einfügen und Löschen von Zeichen / Feldern / gesamtem Schirm; Kopieren von Feldern; Überprüfen von Feldinhalten (Bereichsüberprüfung; verlangt /

nicht verlangt); Anzeigen von Fehlermeldungen, wenn Fehler aufgetreten sind. Automatischer Transfer aller Feldinhalte von Feldern mit read-only- oder read-write-Attributen in die entsprechenden Variablen, nachdem die gesamte Eingabe fehlerfrei abgeschlossen wurde.

Diese extreme Flexibilität hat zur Folge, daß in vielen Fällen Wartungsarbeiten an den Programmen vorgenommen werden können, ohne daß neuerliche Kompilation notwendig ist. Für Standard-Anwender-Software ergibt sich darüber hinaus der Vorteil, daß verschiedene Versionen eines Softwaresystems bestehen können. Solche "individuelle" Systeme werden in exzellenter Weise von DPF unterstützt: das eigentliche Programm ist unveränderbar; die Maskenbibliothek jedoch kann individuell angepaßt werden, wobei diese individuelle Anpassung infolge der Mächtigkeit von DPF sehr weit gehen kann.

3 Das DPF Support System

3.1 Grundlagen

Masken und Druckerformulare, im folgenden kurz immer nur als "Masken" bezeichnet, bestehen aus drei Teilen:

- Hintergrundinformation: sie ist der Teil der Maske, der unverändert bleibt, wann immer die Maske am Schirm oder Printer erscheint.

- Felder: hier werden Werte von Variablen angezeigt und hier können Daten eingegeben werden.

- Attribute der Felder: dies sind Zusatzinformationen über die Felder.

Masken werden mit Maskennummern angesprochen, Felder mit Feldnummern. Es können gleichzeitig mehrere Masken am Schirm stehen, jedoch wird immer nur eine Maske gleichzeitig bearbeitet

(welche, bestimmt das Programm). Zwischen Feldern auf der Maske und Variablen des Programms wird über eine Nummer eine Verbindung hergestellt (im allgemeinen einmal bei Programmbeginn). Der gesamte Datenverkehr zwischen dem Programm und den Masken wird vom Programmierer nur noch durch Aufrufe an die Prozeduren READT (read-terminal) und WRITET (write_terminal) bewerkstelligt. Der Datentransfer aus den Feldern in die Variablen und umgekehrt erfolt vollautomatisch. Abgesehen von der Annehmlichkeit der kompakten Schreibweise ermöglicht dies auch tiefgreifende Änderungen im Programm, ohne daß eine einzige Zeile im Code geändert werden muß. Es braucht nur die Maskendefinition geändertzuwerden, etwa durch Hereinnahme von zusätzlichen Variablen, oder durch Elimination von einzelnen Variablen.

3.2 Das Maskenverwaltungsystem

Eine Maskenbibliothek kann als Datenbank angesehen werden. Das Verwaltungssystem untersützt dementsprechend die Funktionen

-- Einspeichern von neuen Masken
-- Löschen von alten Masken
-- Abändern von bestehenden Masken
-- Informationen über die gesamte Bibliothek und einzelne Masken

Der Maskengenerator liest Maskendeklarationen ein, prüft deren Gültigkeit, übersetzt korrekte Masken in eine interne Darstellung und speichert diese in einer Maskenbibliothek ab. Wesentlich hierbei ist, daß die Masken als gewöhnliche Textfiles mit dem normalen Betriebssystem-Editor erstellt werden können und daher unproblematisch auf andere Rechner und Betriebssysteme gebracht werden können.

3.3 Feldattribute

Folgene Attributarten werden unterstützt:

```
-- Datentypen

   boolean
   integer, long_integer, short_integer
   float, long_float
   character
   picture

-- I/O Attribut

   RO read_only
   RW read_write
   WO write_only

-- Bereichseinschränkungen

   Es kann für ein Feld ein Wertebereich definiert werden,  z.B.
   3..7 oder "AAAA".."HHHH" usw.

-- Zusatzattribute

   noecho       Zeichen werden am Schirm nicht ausgegeben
   map          Kleinbuchstaben werden bei der  Eingabe
                automatisch in Großbuchstaben verwandelt
   blink        Felder blinken am Schirm
   revers       Felder werden "invers-video" dargestellt
   underline    Feldinhalte werden unterstrichen
   click        Hörbares Geräusch bei Eingabe zu diesem Feld
   xchar        jedes  Zeichen ungleich Blank  wird   als   "x"
                ausgegeben ("Ankreuzfeld")
   zeroblank    Die Zahl "0" wird als Blank dargestellt
   zerofill     Alle führenden Blanks werden als "0" dargestellt
```

3.4 Das Support Package

Im Support Package sind jene Prozeduren, Funktionen und Daten zusammengefaßt (eingekapselt), die vom Programmierer zur Bedienung des Systems in einem Anwenderprogramm verwendet wer-

den. Es gibt vier Bereiche:

-- Initialisierung und Beendigung von DPF

-- Gemeinsamer Bereich für Bildschirm und Drucker

   Öffnen und Schließen von Maskenbibliotheken
   Verbinden von Variablen mit Feldnummern
-- Bildschirmbereich

   Initialisierung des Bildschirms
   Datentransfer
      Maskenanzeige am Schirm (bis zu 10 gleichzeitig)
      Ausgabe der Variablen einer Maske am Schirm
      Einlesen der Variablen einer Maske vom Schirm
   Hilfsfunktionen
      Schreiben einer programmspezifischen Fehlermeldung
      Schreiben eines Hintergrundtextes
      Bestimmen eines angekreuzten Feldes in einer Liste
      Löschen aller Felder einer Maske

-- Druckerbereich

   Ausgabe einer Maske am Drucker
   Ausgabe einer Maske auf ein File
   Ausgabe einer Maske in einen Speicherbereich

3.5 Wartung der hardwarespezifischen Information

Alle hardwarespezifische Information ist getrennt von der Maskenbibliothek und getrennt vom Programm auf einem eigenen File gespeichert. Dies ermöglicht es, Hardwarecharkteristika zu ändern, ohne daß das Programm oder die Masken neu übersetzt werden müssen. Für Standardgeräte (d.h. für die meisten handelsüblichen Bildschirme und Drucker) gibt es spezifische Files. Zusätzlich dazu gibt es ein interaktives Wartungsprogramm, das jedes einzelne Hardwareattribut abändern kann.

4 Erfahrungen

4.1 Erfahrungen bei der Entwicklung von DPF

Als sehr günstig erwies sich, daß sowohl das Design von DPF als auch die Programmentwicklung von Personen gemacht wurde, die selbst reichhaltige Erfahrung bei der Entwicklung von interaktiver Anwendersoftware hatten. DPF wurde bereits in der Entwicklungsphase in der Anwenderprogrammierung eingesetzt. Dadurch konnten frühzeitig Fehler bei einzelnen Designpunkten erkannt und ausgemerzt werden. Dies hat zur Folge, daß DPF nicht nur im Gesamtkonzept, sondern auch in den einzelnen Details ausgereift ist und den tatsächlichen Bedürfnissen gerecht wird.

4.2 Erfahrungen beim Einsatz von Anwendersoftware, die DPF als Mensch-Maschine-Schnittstelle verwendet

Es ist zwar nicht das Ziel der vorliegenden Arbeit, auf die ergonomischen Aspekte von Software-Systemen aus der Benutzersicht einzugehen. Es konnte aber festgestellt werden, daß bereits kleine Designfehler beim Maskenaufbau oder Programmablauf zur Ablehnung durch die Benutzer führen können. In diesem Zusammenhang erweist sich die Flexibilität von DPF als vorteilhaft, die notwendige Änderungen schnell und in einfachster Weise zuläßt. Die Schwierigkeit, verschiedene Benutzergruppen zufriedenzustellen (naiver Benutzer / Experte; sporadische Benutzung / permanente Benutzung), ein Problem, auf das in der Literatur immer wieder hingewiesen wird, kann von DPF ausgezeichnet bewältigt werden. Wir konnten diese Schwierigkeit leicht umgehen, indem wir z.B. verschiedene Maskenbibliotheken bereitstellten mit kompakter Information für den geübten Benutzer und ausführlicher Information und Menüsteuerung für den ungeübten Benutzer.

DPF wird derzeit in zwei großen Standard-Anwender-Software-Systemen verwendet (Büroinformationssystem für Rechtsanwaltskanzleien und Inkassobüros), einem Studenteninformationssystem, einem Unternehmensspiel zur Beschaffung und Lagerhaltung und mehreren kleineren und mittleren Anwenderprogrammen.

DPF wird für gerade in Entwicklung stehende Softwaresysteme eingesetzt werden: für ein relationales Datenbanksystem, ein statistisches Analysesystem und für ein ADA Simulation Support Environment.

5 Literaturverzeichnis

/1/ Adelsberger, H. H.; Rockenschaub, H.: Supporting Systems for Software Development. Areitsberichte zum Tätigkeitsfeld Wirtschaftsinformatik, Wirtschaftsuniversität Wien, #5/80, Wien 1980

/2/ Adelsberger, H. H.; Wagner, M.: DPF: A Uniform Screen and Printer Input-Output System. Department of Applied Statistics and Data Processing, Technical Report #82/10, Wien 1982

/3/ Galitz, W. O.; Handbook of Screen Format Design. Q.E.D. Information Sciences, Inc., Wellesley, MA

/4/ Heffler, M. J.: Description of a Menu Creation and Interpretation System. Software-Prac. & Exp. 12 (1982) 269 - 281

/5/ Ledgard, H.; Singer, A.; Whiteside, J.: Directions in Human Factors in Computer Information Systems. Lecture Notes in Computer Science, No. 103, Springer-Verlag, New York, 1981

/6/ Thomas, J. C.; Caroll, J. M.: Human factors in communication. IBM Syst.J. 20, 2 (81) 237-263

Dr. Heimo H. Adelsberger
Wirtschaftsuniversität Wien
Augasse 2-6

A - 1090 W i e n, Austria